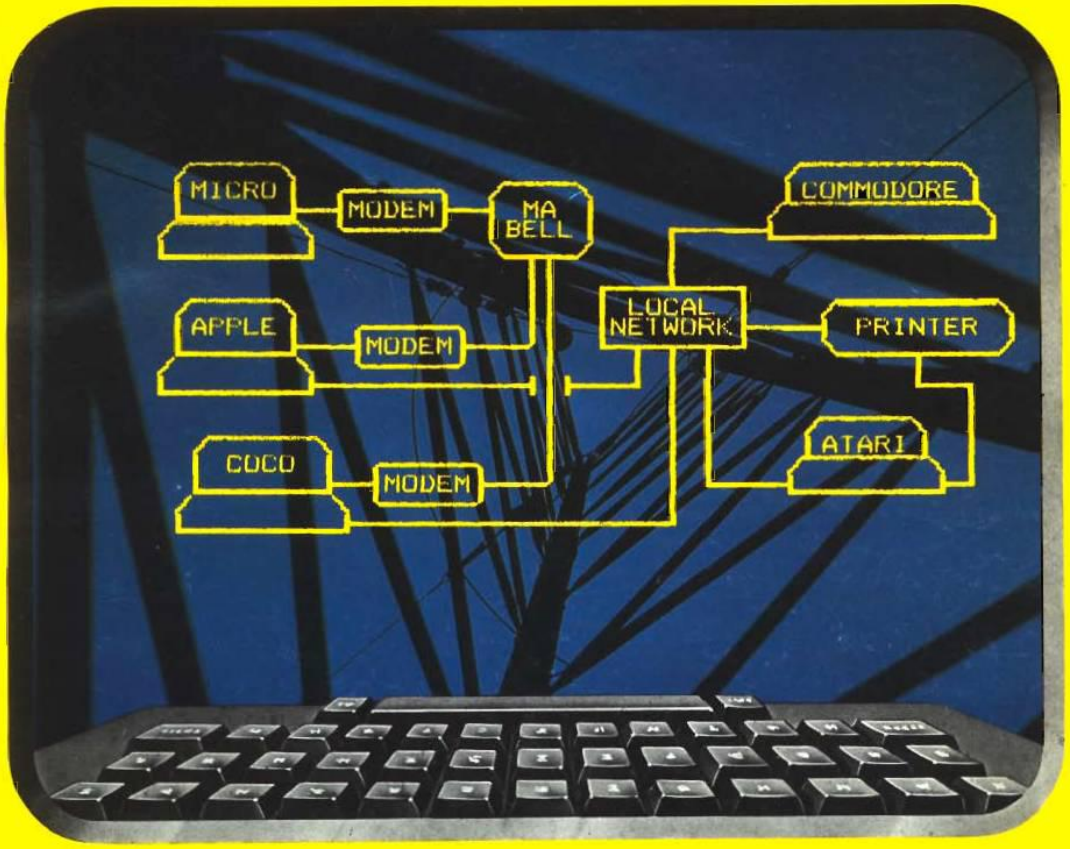
NO. 68 COMMUNICATION
U.S. Edition: $2.50
International Edition: $3.00

Complete Programs Included

MICRO™

for the Serious Computerist

COMMUNICATION



Smart Modem Program
for the ● Apple ● Atari ● C-64 ● CoCo

MODCOMM: A Simple Modem Driver for CoCo

DOSPLUS: New Commands for Commodore 64

SPEEDUP: Reduce Apple Disk Loading Time

Atari Player: Electronic Organ Program

01
0 74470 11690 0

Richvale Telecommunications

10610 BAYVIEW (Bayview Plaza)
RICHMOND HILL, ONTARIO, CANADA L4C 3N8
(416) 884-4165

$185.00 Canadian
$149.00 U.S.
PLUS CUSTOMS BROKERAGE, HANDLING AND MAILING CHARGE.

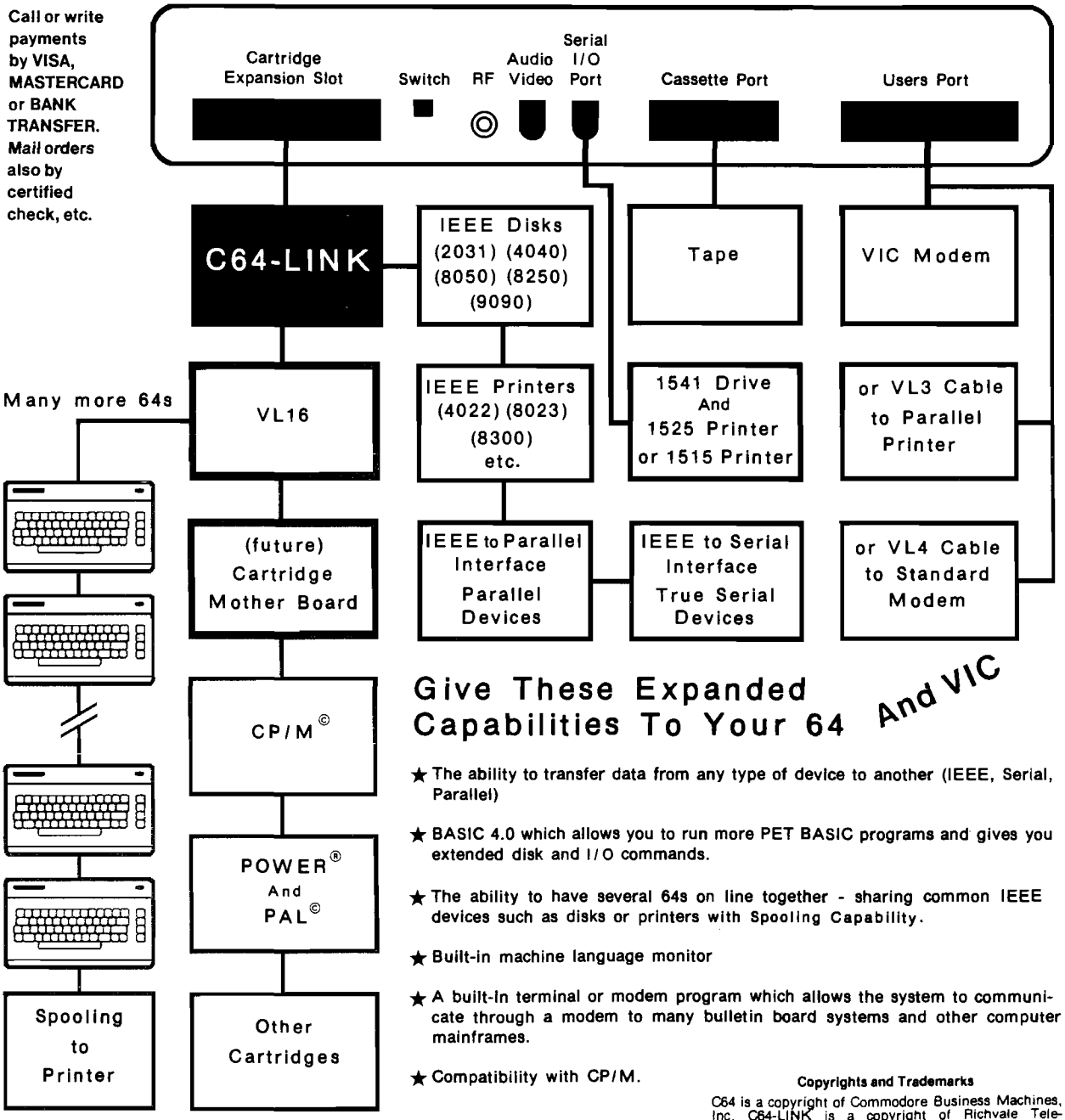
RTC

C64-LINK©
The Smart 64

Also available for VIC 20

RTC

Call or write
payments
by VISA,
MASTERCARD
or BANK
TRANSFER.
Mail orders
also by
certified
check, etc.



Give These Expanded Capabilities To Your 64 And VIC

★ The ability to transfer data from any type of device to another (IEEE, Serial, Parallel)

★ BASIC 4.0 which allows you to run more PET BASIC programs and gives you extended disk and I/O commands.

★ The ability to have several 64s on line together - sharing common IEEE devices such as disks or printers with Spooling Capability.

★ Built-in machine language monitor

★ A built-in terminal or modem program which allows the system to communicate through a modem to many bulletin board systems and other computer mainframes.

★ Compatibility with CP/M.

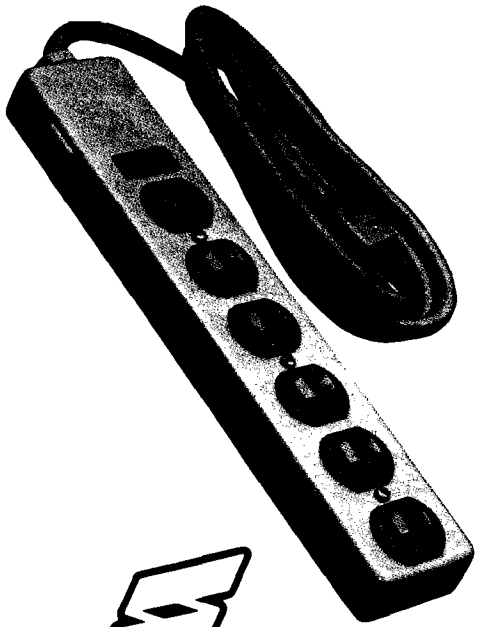
Contact your local Commodore dealer or RTC.

Copyrights and Trademarks

C64 is a copyright of Commodore Business Machines, Inc. C64-LINK is a copyright of Richvale Telecommunications. CP/M is a registered trademark of Digital Research. POWER is a trademark of Professional Software. PAL is a copyright of Brad Templeton.

another FLEXIDUCT® OFFICE SAFETY PRODUCT

When you least expect it, ZAP!!!



In a few millionths of a second, common electrical surges and spikes can enter your data processing equipment and cause memory loss, false logic and misregistration. Surges very often do permanent damage to microcircuitry.

FLEXIDUCT Surge Suppressors catch surges and spikes before they have a chance to enter your equipment. In billionths of a second (Nanoseconds),**FLEXIDUCT** Surge Suppressors dissipate surges and spikes from any side of the line (most protect only one side).

The multi-outlet unit is ideally suited to the computerized work-station. It gives you the advantage of circuit breaker protection with plenty of outlets for data processor and peripheral equipment.

No computer should be without the protection of a **FLEXIDUCT** Surge Suppressor...**especially yours!** Write or call for further information. Available from office products retailers.

FLEXIDUCT® Surge Suppressors

a product of Winders & Geist, Inc. P.O. Box 83088 Lincoln, NE 68501 402/474-3400

MICRO™

Editorial

Communications

The focus of this issue of **MICRO** is on communications: computer-to-computer over telephone modems -including program downloading and computer bulletin boards, and computer-to-computer communication *via* local networks.

Remote Communications

An original article by John Kelty on **CoCo and the Hayes Smart Modem** has been modified by Phil Daley to work on the Apple, Atari, and Commodore 64 as well. No matter which computer you own, you now have the software you need to run this popular modem. Walter Charlton has taken a different approach to add communications to his CoCo. **MODCONN: The CoCo and Modem-to-Modem mmunication** shows how to add RS-232 capability without fancy hardware.On the Commodore computers you have to use some tricks to download BASIC. Stone and Cornwall show how this may be done on the PET, VIC and C64. Part of Paul Swanson's column deals with **Telecommunicating with your Atari** and the use of free bulletin boards.

Local Networks

Randall Hyde provides a comprehensive introduction to local networking with in **The Network Primer**. He describes the seven-layered network, discusses several important network topologies, and then shows how these concepts are involved in one particular system, the NESTAR CLUSTER ONE.

MICRO Communications

One of the reasons for featuring communications at this time is that we are developing a **MICRO Phone Service** for subscribers. In addition to being a general purpose bulletin board for serious computerists, authors will be able to submit articles and programs, and subscribers will be able to download programs that have appeared in MICRO. Printed listings have always been a problem. The longer the program listing, the greater the chance of making typing errors. Once we have the **MICRO Phone Service** running, the only effect of length will be the cost of telephone connect time! We will have to establish some protocols for program loading, standards for assembler formats, and so forth. I would appreciate hearing from those of you who have had experience in this area.

Robert M. Tripp

Editor-in-Chief

DISCOUNT COMPUTER SOFTWARE ACCESSORIES

APPLE

	Retail	Discount
Eliminator	$29.95	21.00
War	24.95	18.00
Adventureland	29.95	21.00
Pirates Adventure	29.95	21.00
Golden Voyage	29.95	21.00
Magic Window	99.95	72.00
Temple of Apshai	39.95	29.00
Upper Reaches of Apshai	19.95	15.00
Curse of Ra	19.95	15.00
Midway Campaign	16.00	12.00
Hi-Res Computer Golf	29.95	21.00
DOS Boss	24.00	18.00
The Arcade Machine	44.95	33.00
Star Blazer	31.95	23.00
Choplifter	34.95	25.00
Serpentine	34.95	25.00
Deadly Secrets	34.95	25.00
Raster Blaster	29.95	21.00
Bug Attack	29.95	21.00
The Home Accountant	74.95	54.00
Snack Attack	29.95	21.00
Pig Pen	29.95	21.00
Wordrace	24.95	18.00
Rendevous	39.95	29.00
Russki Duck	34.95	25.00
Horizon V	34.95	25.00
Sargon II	34.95	25.00
Zork I	39.95	29.00
Zork II	39.95	29.00
Deadline	49.95	36.00
Mastertype	39.95	29.00
Castle Wolfenstein	29.95	21.00
Supertext II	150.00	108.00
Softcard Premium System	775.00	600.00
Wizard and the Princess	32.95	24.00
Time Zone	99.95	72.00
Cranston Manor	34.95	25.00
Threshold	39.95	29.00
Softporn Adventure	29.95	21.00
Crossfire	29.95	21.00
Frogger	34.95	25.00
Laff Pak	34.95	25.00
Ultima II	59.95	44.00
Screenwriter II	129.95	94.00
Graphics Magician	59.95	44.00
Pie Man	29.95	21.00
Fastgammon	24.95	18.00
Congo	34.95	25.00
Goldrush	34.95	25.00
Gorgon	39.95	29.00
Beer Run	29.95	21.00
Snake Byte	29.95	21.00

Intec 32K Board $75.00
APPLE Compatible Disk Drive $265.00
VERBATIM/DATALIFE Disks $26.00

SPECIAL OFFERS

ATARI

	Retail	Discount
Threshold (d)	$39.95	29.00
Snake Byte (d)	29.95	21.00
Space Eggs (d)	29.95	21.00
Bandits (d)	34.95	29.00
Color Print (d)	39.95	29.00
Canyon Climber (d)	29.95	21.00
Shooting Arcade (d) (t)	29.95	21.00
Pacific Coast Highway (d) (t)	29.95	21.00
Clowns And Balloons (d) (t)	29.95	21.00
Wordrace (d)	24.95	18.00
Andromeda (d)	34.95	25.00
Deadline (d)	49.95	36.00
Zork I (d)	39.95	29.00
Zork II (d)	39.95	29.00
Alien Swarm (d)	34.95	25.00
Action Quest (d) (t)	29.95	21.00
Ghost Encounters (d) (t)	29.95	21.00
K-Razy Shootout (c)	49.95	36.00
K-Razy Kritters (c)	49.95	36.00
Ultima I (d)	39.95	29.00
Ali Baba and Forty Thieves (d)	32.95	24.00
Deluxe Invaders (c)	39.95	29.00
Gorf (c)	49.95	36.00
Wizard of Wor (c)	49.95	36.00
Preppie (d) (t)	29.95	21.00
Tigers in The Snow (d) (t)	39.95	29.00
Ghostly Manor (d)	24.95	18.00
Raster Blaster (d)	29.95	21.00

T=Cassette
D=Disk
C=Cartridge

	Retail	Discount
Rear Guard (d)	24.95	18.00
Rear Guard (t)	19.95	15.00
Caverns of Mars (d)	39.95	29.00
Atari Basic (c)	59.95	45.00
Star Raiders (c)	44.95	33.00
Centipede (c)	44.95	33.00
Pac Man (c)	44.95	33.00
Pilot (c)	79.95	60.00
Temple of Apshai (d) (t)	39.95	29.00
Upper Reaches of Apshai (t)	19.95	15.00
Curse of Ra (d)	19.95	15.00
Midway Campaign (t)	16.00	12.00
Apple Panic (d)	29.95	21.00
Track Attack (d)	29.95	21.00
Choplifter (d)	34.95	25.00
Star Blazer (d)	31.95	24.00
Wizard and the Princess (d)	32.95	24.00
Jawbreaker (d) (t)	29.95	21.00
Crossfire (d) (t)	29.95	21.00
Frogger (d) (t)	34.95	25.00
The Shattered Alliance (d)	39.95	29.00
Battle of Shiloh (d)	39.95	29.00
Submarine Commander (c)	49.95	39.00

MANY MORE PROGRAMS AVAILABLE — VISA AND MASTERCARD ACCEPTED

TERMS: Send check or money order for total purchase price, plus $2.00 for shipping. MI residents add 4% tax. C.O.D. accepted.

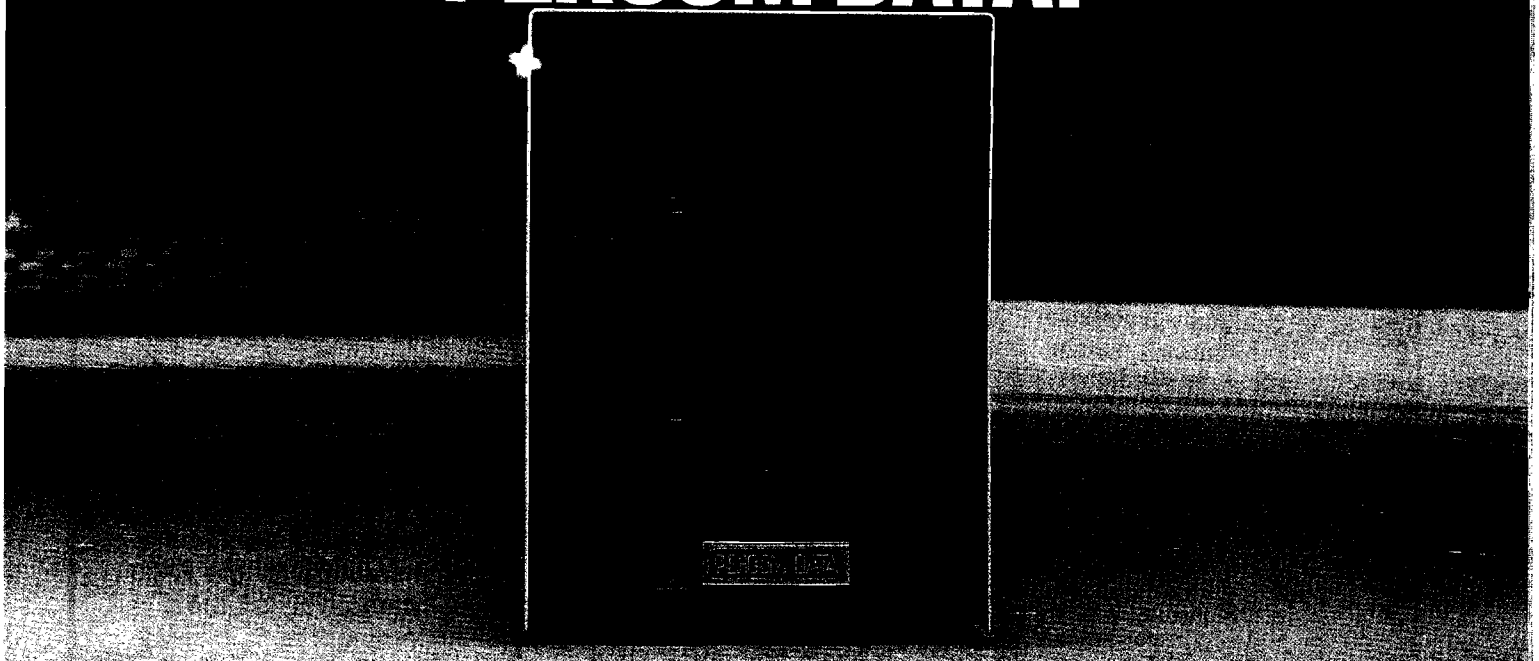
® MFGS TRADEMARK

STRÖM SYSTEMS INC.
P.O. Box 197
Plymouth, Mi. 48170
(313) 455-8022

WRITE OR CALL FOR FREE CATALOG
PHONE ORDER HOURS
4 PM - 9 PM MON. - FRI.
INCLUDE CARD NUMBER AND EXPIRATION DATE WITH CREDIT CARD ORDERS.
INCLUDE TYPE OF COMPUTER.

ATARI COMPUTER OWNERS:

Pick the positively perfect, practical, peripheral package, from PERCOM DATA!



That's right... the positively perfect PERCOM DATA 5¼" floppy disk drive with a BUILT-IN PRINTER-PORT, for your Atari® 400/800 is now available!

Until now, Atari computer owners who wanted to hook a printer to their computer had only one choice... spend about $220 for an interface device. THOSE DAYS ARE OVER. PERCOM DATA has built a parallel printer-port right into its new AT88 PD model. Now you can add a quality disk drive system AND have a place to plug in a printer... WITHOUT BUYING an interface.

The AT88 S1 PD™ disk drive operates in both single density (88K bytes formatted) and double density (176K bytes formatted).

What more could you want? NO INTERFACE... a high quality PERCOM DATA disk drive... AND a built-in PRINTER-PORT... all with a price of $599.

Pick up a positively perfect PERCOM DATA disk drive, with printer-port... pronto!

For the name of an authorized PERCOM DATA Dealer near you, call our **TOLL-FREE HOTLINE 1-800-527-1222 NOW**, or write for more information.

Perfectly Priced

$599.

PERCOM DATA
CORPORATION

Expanding Your Peripheral Vision

DRIVES • NETWORKS • SOFTWARE

11220 Pagemill Road, Dallas, Texas 75243 (214) 340-5800
1-800-527-1222

Atari is a registered trademark of Atari, Inc. • AT88 S1 PD is a trademark of Percom Data Corporation • COPYRIGHT PERCOM DATA CORPORATION 1983
Prices subject to change without notice

MICRO™

for the ***Serious Computerist***

MICRO
P.O. Box 6502
Chelmsford, MA 01824
617/256-3649

Publisher/Editor-in-Chief
Robert M. Tripp

Associate Publisher
Cindy Kocher

Technical Editor
Phil Daley

Production Manager
Nancy Lapointe

Typesetter
Lynda Fedas

Advertising Manager
Cindy Kocher

Sales Manager
C. Skip Bentle

Circulation Manager
Linda Hensdill

Customer Service
Kim Dundas

Accounting
Donna M. Tripp

Contributing Editors
Cornelis Bongers
David Malmberg
John Steiner
Jim Strasma
Paul Swanson
Richard C. Vile, Jr.
Loren Wright

MICRO is published monthly by:
MICRO, Chelmsford, MA 01824.
Second Class postage paid at:
Chelmsford, MA 01824 and
additional mailing offices.
USPS Publication Number: 483470.
ISSN: 0271-9002.
Send subscriptions, change of address, USPS Form 3579, requests for back issues and all other fulfillment questions to:
MICRO
P.O.Box 6502
Chelmsford, MA 01824
or call 617/256-3649.

Subscription Rates: (per year):

U.S. $24.00 or $42.00 for two years
Foreign surface mail: $27.00
Air mail: Europe $42.00
Mexico, Central America, Middle East, North Africa, Central Africa $48.00
South America, South Africa, Far East, Australia, New Zealand $72.00

Copyright © 1983 by MICRO.
All Rights Reserved.

Communication

7 The Network Primer
Randall Hyde
The organization and implementation of local networks

Using the Smart Modem in Your System:
John Kelty
A BASIC program that services the Hayes Smart Modem for the following microcomputers:
Apple
Atari
Color Computer
Commodore 64

62 MODCOMM
Walter Charlton
A simple modem to modem communication program for the Color Computer

17 Downloading BASIC on PET, VIC and C64
Kevin Stone & Andrew Cornwall
A program to allow BASIC to be downloaded over a modem into the family of Commodore computers

Commodore

17 Downloading BASIC on PET, VIC and C64
Kevin Stone & Andrew Cornwall
A program to allow BASIC to be downloaded over a modem

22 DOSPLUS for Commodore 64
Michael Keryan
Add new utility functions called from the keyboard

33 Generating Characters for the EPSON FX80 on the Commodore
Robert M. Tripp
Create user defineable characters for the FX80 interactively on the Commodore 64 display

39 Commodore Reviews

40 Commodore to Smart Modem
John Kelty & Phil Daley
A BASIC program that services the Hayes Smart Modem for the C64

43 Commodore Compass
Loren Wright
High resolution graphics on the Commodore 64

Color Computer

60 CoCo to Hayes Smart Modem
John Kelty & Phil Daley
A BASIC program that services the Hayes Smart Modem for the Color Computer

62 MODCOMM
Walter Charlton
A simple modem to modem communication program for the Color Computer

67 A VOX Input for Your CoCo
Sean Moyer
Subroutines that allow sound to be used as input to the computer

69 CoCo Bits
John Steiner
The OS-9 operating system

71 TRS-80C Reviews

Atari

46 Atari to Hayes Smart Modem
John Kelty & Phil Daley
A BASIC program that services the Hayes Smart Modem for the Atari

49 Atari Music Player
Tom Marshall
Compose, save, load and replay songs from your Atari keyboard

57 From Here to Atari
Paul Swanson
The ATR8000 peripheral controller and Bulletin Boards

59 Atari Reviews

General

7 The Network Primer
Randall Hyde
A primer describing the fundamentals of local networks, with specific examples on the NESTAR system

91 Interface Clinic
Ralph Tenny
Factors involved in getting two or more microprocessors to work together

Apple

74 Apple to Hayes Smart Modem
John Kelty & Phil Daley
A BASIC program that services the Hayes Smart Modem for the Apple

79 Speedload: Enhanced Apple DOS
Enirco Colombini
An enhanced version of DSO 3.3 that performs loading operations about four times faster.

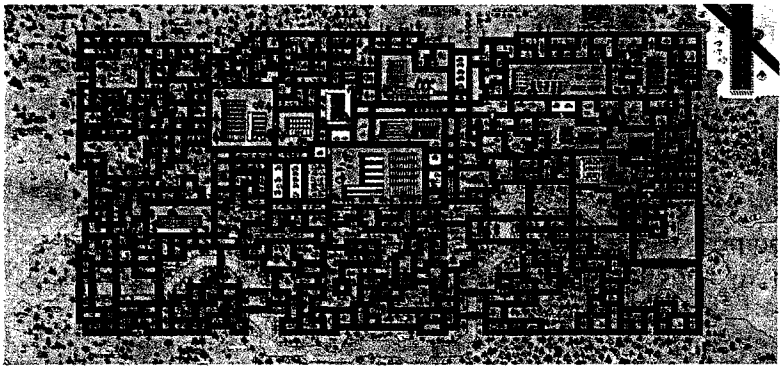
71 Apple Reviews

DEPARTMENTS

2 Editorial
112 Advertiser's Index

GETAWAY!

You've got the loot ... now, *GETAWAY!* to a great new game from the Atari® Program Exchange!



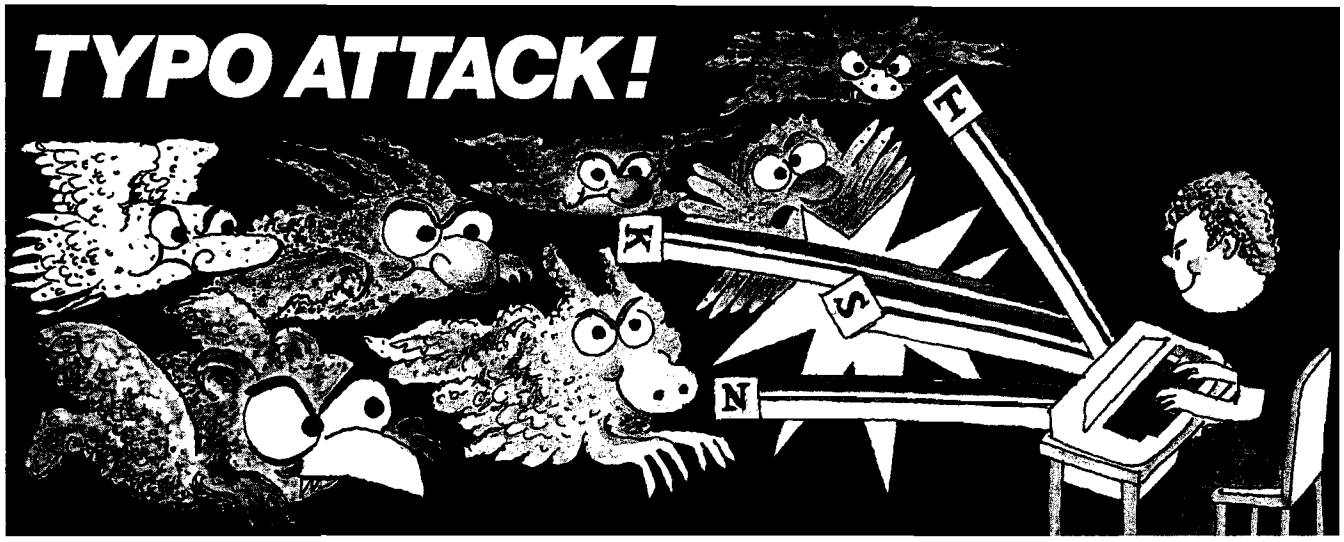
It's *all* there! The squall of sirens, the crazy turns down endless city streets, the anxious search for ill-gotten gain, the race against time for a safe place to stash your cash! Now your gas tank is nearly empty and night is about to fall. The coppers are closing in fast. Before you learn again that crime doesn't pay...Quick! ***GETAWAY!***

Ask for *GETAWAY!* at your local Atari software retailer, or order direct. Phone 800-538-1862, or 800-672-1850 in California. Or write Atari Program Exchange, P.O. Box 3705, Santa Clara, CA 95055.

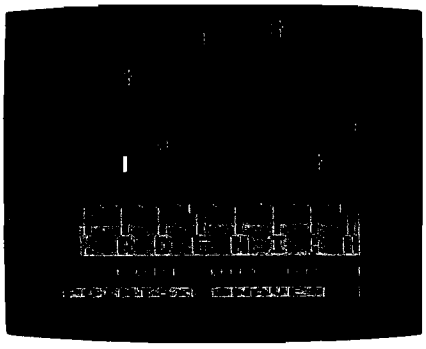
Cassette (410)	APX-10195 32K	$29.95
Diskette (810)	APX-20195 32K	$29.95

For direct orders, add $2.50.

TYPO ATTACK!



You're in for a nasty spell ... unless you stop the Typos!



In the dusky world beneath your keyboard the gruesome Typos dwell ...waiting to attack! Term paper due tomorrow? Got to get that book report typed? Fool! The Typos will devour your letters as you type! That could spell D-O-O-M-E for *you!*

Before you start typing, get down to the real work: destroy the Typos before they destroy your proase... uh, proze...prrrose...Oh NO! THE TYPOS!!! Get ***TYPO ATTACK***, a grand and glorious game from Atari® Program Exchange. It might even improve your typing!

Ask for ***TYPO ATTACK*** at your local Atari software retailer, or order direct. Phone 800-538-1862, or 800-672-1850 in California. Or write Atari Program Exchange, P.O. Box 3705, Santa Clara, CA 95055.

Cassette (410):	APX-10180	8K	$29.95
Diskette (810):	APX-20180	16K	$29.95

For direct orders, add $2.50.

The Network Primer

by Randall Hyde

In the early sixties the integrated circuit was introduced. Engineers were delighted to pay $20.00 for a single DIP package containing four NAND gates. Due to the expense involved, every device designed using these devices was optimized to use as few of these devices as possible. With every technological advance that came along the high price of silicon dropped dramatically. Today you can buy that same NAND gate for less than fifteen cents in suitable quantities.

Since computers designed in the sixties required thousands of those $20.00 NAND chips, computers (specifically CPU's) designed at that time were very expensive. Since CPU's were so expensive computer center administrators wanted to squeeze every bit of useful computational power out of their system as was possible. It was considered heresy to leave a computer idle. A million-dollar instrument sitting around idle was a very expensive waste of money.

The same advances in silicon technology that gave us the fifteen cent NAND gate also gave us the microprocessor. It is now possible to buy a complete computer for less than $2.00 (Intel just announced an 8048-type microcomputer for less than $2.00 containing CPU, RAM, ROM, and I/O — everything you need for a complete computer system). Even considering a personal computer, such as the Apple II system, that sells for $1,000 to $2,000 it becomes apparent that the CPU utilization is not very important — if the computer sits around all day with a blinking cursor waiting for input from the user, no big deal, for the same million dollars that bought the System/360 in 1965 you could buy 500 to 1,000 personal computers and get more through-put even if over three-quarters of the computers sat idle at any given time.

There were only a few problems associated with using 1,000 personal computers instead of a single, large computer. First, although the overall through-put is increased tremendously, certain compute-bound operations such as math-intensive floating point calculations will proceed much slower on a personal computer than on a larger computer complemented with a floating point instruction set. Second, while the price of the CPU has dropped dramatically, electromechanical devices such as printers, disks, tape drives, and other such devices haven't dropped in price by the same proportion (i.e., these devices are still relatively expensive). Lastly, with the big expensive computers several users and processes could communicate with one another and share databases. With the localized personal computer it is difficult, if not impossible, for different users to share a common database or communicate with one another. For this reason many computer departments continue to purchase large computers instead of decentralized personal computers.

The need to share peripheral devices is, of course, very important. While a single user may have the need to use a printer, he might not be able to justify the high cost of such a device for use at his work station alone, especially if it is a special purpose printer such as a letter quality printer that will be used infrequently. It would be much better if a lot of different users could have access to a single letter-quality printer. Since each user would only be using the letter quality printer a small fraction of the time, it would almost always be available for use whenever a user desired to print something on it. Since the printer is available to several users, instead of just one, its utilization would be much higher than if it were connected to a single work station. Using the same idea hard disks, real-time clocks, floating point processors, and other special devices would be much more cost effective if they could be shared amongst several users.

Several manufactures have created special multiplexing devices that allow several personal computer users to share a particular hardware device. For example, CORVUS has a device called the CORVUS Constellation that lets several microcomputer systems share a CORVUS hard disk system. Other manufacturers provide black boxes that let several personal computer users share printers, tape drives, and other peripheral devices. While this approach lets personal computer users share peripheral devices (and to a lesser extent, share databases), it suffers from the drawback that it does not allow users to easily communicate with one another, nor does it let users share exotic peripherals for which there is no multiplexer available.

The Answer: A Network

To solve the problems inherent with a bunch of autonomous computers a technique employed by larger computers is employed: the computer network. A computer network is simply some medium that allows two or more computers to communicate with one another. In a very general sense, the RS-232 line is an example of a computer network since it allows computers to communicate with one another. Real-life computer networks usually use a different transmission scheme than RS-232, but the idea is exactly the same-some method for transmitting data between two computers is used.

In general there are two types of computer networks: Remote networks and Local networks. A remote network is usually associated with large mainframe computers. Anyone who has connected to the SOURCE via TELENET or some other service has used a remote networking system. Remote networks are seldomly used as a means to share peripherals such as printers, hard disks, etc. since data transmission rates are very slow and the end user typically has little access to such device (so the hard copy output, for example, could be picked up). Remote networks do allow access to shared data bases such as the UPI and API data bases and stock quotation from the DOW JONES and other financial systems. In general, however, remote networks are of little interest to the group of personal computer users who wish to share a common set of peripheral devices and files.

The local network is used to connect various devices that are located within a short distance of one another. For example, within an office building you might place a fast dot-matrix printer in the engineering lab, in the accounting department, in the sales department, and in the shipping departments. Persons using a local network would send draft output to the printer nearest their workstation. A single letter-quality printer might be found in the sales department. Anyone could use the letter-quality printer by sending the data to the proper device. The choice of dot-matrix printer would be made simply by distance to the printer. The people in the engineering lab would probably use the printer in the engineering lab since it is closer (although if the printer in the engineering lab is tied up for a while output could be directed to the sales department's printer).

The local network also lets several individuals share a single high-speed, high-capacity hard disk. This also allows several users to share a common database and other files. For special applications such as floating point operations, a special floating point processor can be placed on the network and complex floating point operations can be loaded into the floating point processor for further calculations.

Finally, a local network lets individual users communicate with one another. Since, after all, a network is simply a medium of communication between two or more computers, two users can easily communicate by sending data from computer to computer. This allows conversations to take place "CB" style over the network.

Within a local network there are two types of users: clients and servers. A server is a computer system on the network (a computer system on the network is called a node) that provides a service to other nodes. For example, a file server may consist of a small computer system connected to a hard disk. It recognizes requests from other users in the system and reads data from (or writes data to) the hard disk. A client is some node that requests service from a server. For example, a typical user requesting data from the file server is a client of the file server. Note that a node can be both a server and a client. For example, the file server may become a client of a node containing a real-time clock in order to obtain information to "date-stamp" a file.

The Seven-Layered Network

Most of the modern network technology is directly attributable to researched performed by the Department of Defense and several major universities. A few years back the International Standards Organization (ISO) decided to create a draft containing several standards for implementing network systems. A subset of the standard (in one fashion or another) is implemented in almost all modern networking systems. Therefore, an understanding of the ISO model is extremely useful for understanding specific networking systems.

The ISO model divides the basic networking system into seven major components or layers. Each successive layer builds on the layer just below it (see figure 1). The physical layer is the electrical and mechanical specifications of the networking system. For example, consider the RS-232 standard. Mechanically, a DB-25 connector is used with certain pins allocated certain functions. This is the

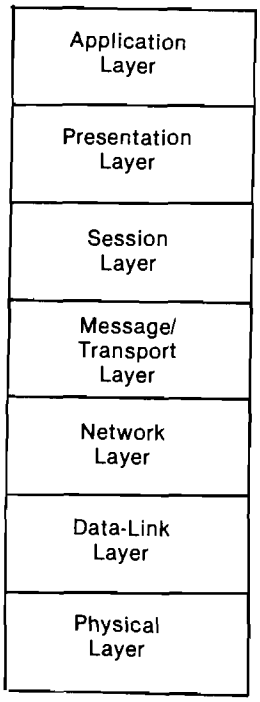


Figure 1: **ISO seven-layer network model.**

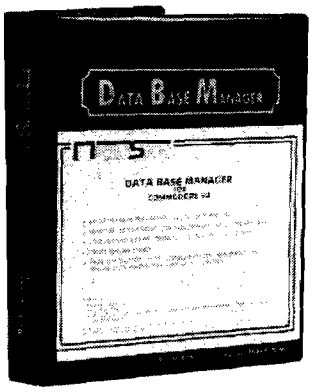
mechanical specification of the RS-232 standard. Electrically, the signal is allowed to vary between 0v and + 12v for a logic one; and between 0v and – 12v for a logical zero. These specifications are the electrical specs for the RS-232 system (granted, somewhat abbreviated). The Physical Layer for any network system defines the connection in these terms, the connector used, voltage ratings, type of transmission (phase encoded, pure digital, analog, etc.), and other specifications of interest to the engineer wiring the system together.

The Data Link Layer is responsible for the reading and writing of data to the network, error detection (and possibly correction), and the assembly and disassembly of data packets. More specifically, the data link layer is presented with a packet of data containing, in addition to the data to be transmitted, a destination address and other useful information. The data link layer adds a checksum (or other error-detecting scheme) to the end of this packet and transmits the data over the network. Note that the packet was given to the data link layer as a complete package, the data link layer did not have to assemble the packet a byte at a time. Likewise, when the data link layer reads data from the network it reconstructs the data packet transmitted and presents the higher levels with a complete, error-free packet of data. In the event an error is detected the data link layer disregards the entire transmission and requests a retransmission. The higher layers are completely unaware of this. Most networks provided error-free data above the data link layer.

The Network Layer is usually found only in the remote network systems. The purpose of the network layer to to insure that data packets move smoothly throughout the network system. In most remote network systems data packets being sent from system A to system C must first

MICROSPEC

SOFTWARE MEANS BUSINESS FOR THE COMMODORE 64



When it's time to get serious, it's time to boot up MicroSpec business software. Our complete line of business software is made to give you some real applications for your Commodore 64. From data base management to full accounting software, we have the package for you.

It's attention to detail that makes our packages so beautiful and makes them stand out from the rest. We realize that most people are first time users, so we designed all our packages to be completely menu driven and user prompted for each input. We also know that most people use only one disk drive, so we designed all our packages to virtually eliminate disk swapping. Other features like non destructive input routines really make our software easy to use. But all this doesn't restrict you. Pure random access file structure maximizes your disk capacity and allows you to bring up any record for viewing in less than a second.

In our efforts to put together the best packages available, we worked on more than the software. We took the same approach with the documentation as the software. We made it complete and easily understood for the first time user. We even provide sample reports in many cases.

The Demonstration Package, which shows how each program runs, is available for $19.95. So, if you're serious about your 64, call or write for a complete brochure or **go right down to your nearest computer retailer for a demonstration.**

WHEN YOU AND YOUR 64 ARE READY TO GET DOWN TO BUSINESS
GIVE US A CALL

MICROSPEC

P.O. BOX 863085 • PLANO, TX 75086
(214) 867-1333

pass through system B. Obviously the data being transmitted to point C is of little interest to the application program running on machine B, but due to the structure of the network system B must store and forward the data being received from system A on to system C. The network layer handles this problem as well as other problems associated with congestion and flow control.

The Message/Transport layer concerns itself with machine-machine communications. The transport layer is responsible for taking a complete message and "transporting" it to the destination machine. Likewise this layer is responsible for accepting a message and presenting the whole message to higher layers in the protocol.

The transport layer is responsible for taking a complete message, breaking it up into smaller packets (if necessary), and presenting these packets of data to the network (or data link) layer for transmission. This allows the higher layers to treat data strictly in terms of message units without having to worry about maximum message lengths, minimum messages lengths, the type of transmission, etc.. Another responsibility of the transport layer is handling the type of communication. There are typically two types of transmission "modes":

Virtual Circuits and Datagrams.

Datagrams are very important in remote networks where each data packet can take one of several different routes. In a datagram model the network simply accepts messages and attempts to transport them to the destination as a single unit. Datagrams may arrive out of order or not at all. The receiving station is responsible for collecting these datagrams, assembling them in their proper order, and making sure that all datagrams have arrived and are error-free.

A virtual circuit attempts to treat the network system as though the source and destination stations were the only systems on the network. In this type of setup the network appears as a single circuit between the source and destination stations. Virtual circuits always provide an error-free transmission channel where no data packets are lost and they always arrive in the order they were transmitted in. Once the transmission is set up, all data is transmitted at which point the virtual circuit is disconnected. Most local networks use some combination of virtual circuit/datagram service.

The Session layer is the true user's interface into the network. The session layer is responsible for setting up a communication channel between users and tacking on the necessary address information for each packet transmitted. The session layer is also responsible for making sure the transmission of a message isn't terminated halfway through. For instance, a database management program would be completely destroyed if a session was terminated halfway through the update of a particular record. Typically, the session level would gather all the data required by a particular routine and present the message as a complete package, not allowing the system to continue operation until the entire package was received.

The Presentation layer is typically a small collection of routines that handle data compaction, encryptionation, etc.. This layer is rarely implemented in local networks (or remote networks for that matter).

The Application layer is the user program making use of the network system. The definition of this layer is entirely up to the user.

A Network System You've Used Since Childhood

Believe it or not, you've been using a remote network for a good part of your life. Thanks to Alexander Graham Bell most of the people in this country, even those who've never seen a computer in their life, are well aquainted with the most important network of all: the telephone system. Bell labs (the R & D portion of the Bell system) has put forth considerable research into optimizing the telephone system. Luckily, many of the concepts developed by Bell labs can be directly applied to computer networks. With this in mind, a discussion of the telephone system may be helpful before discussing computer networks.

In many respects, the telephone system can be divided up into the seven layers just described for computer networks. For example, at the physical level you have the connector plug in the wall, the allowable voltages, the number of wires, and (for all you modem fans) esoteric things like the ringer equivalence value. The data link layer is the circuitry required to handle the analog transmission from the mouthpiece to the telephone line. The network layer is responsible for picking out the route which the phone call will take (yes, sometimes a call from L.A. to San Francisco gets routed via Washington D.C.). The transport layer defines a virtual circuit mechanism for data transmission (after all you do want the data to arrive in order, it would be uncool if you said "Hello there" and it came out "there Hello" at the other end). The session layer consists of the circuitry required during dialing and the maintenance of the virtual circuit during communication. The presentation layer (if present) could be represented by one of those magic "black boxes" that the CIA uses to scramble and unscramble phone conversations. And the applications layer is the actual phone conversation taking place. As you can see, the OSI network structure provided by the ISO model is very general and easily re-worded to fit almost any communication system in existence.

It was noted earlier that Bell Labs has done quite a bit of research into network structures, much of which has found its way into many of the modern network designs. This combined with the DOD's ARPANET, IBM's SNA, and other early networking systems has structured many of the modern networking systems including ETHERNET, OMNINET, NESTAR, and others.

An Introduction to Local Networks

The need for local networks grew from two basic desires: wanting to share expensive peripheral devices among several inexpensive CPU's and wanting to improve communication between various workstations.

Consider the average sales office. At any given time you may have several secretaries using word processing stations, executives using some financial modeling system (VISICALC anyone?), salesmen checking stock and placing orders on computer terminals, and clerks maintaining the inventory using workstations in the warehouse. This much activity would bog any conventional timeshare system down tremendously. Could you imagine how

useful VISICALC would be if it took 20 seconds to perform each calculation? So a timeshare system would be out of the question. Yet the individual members have to be able to share data, for example whenever a salesman sold something it would be nice if the stocking clerk was alerting so that he could order a replacement. Likewise, you wouldn't want two salesmen selling the same item if it is the last one in stock, or at least you would want to alert them so that they could advise the customer of any possible delays. By combining a group of small computers into a network it is possible to allow members of a organization to easily share computer data without the hassles of using a timeshare system.

Since there is a need to provide many of the employees with access to some common data (like the number of items in stock) a bunch of isolated microcomputers won't be able to completely solve the problem. But if those same microcomputer systems could be connected together so that peripherals, data, and other communications lines could be shared the problem could be solved. Such wishful thinking lead to the development of the local network.

The ETHERNET networking system is an example of a local network. The ETHERNET system allows office employees to share databases, peripherals, programs, "chat" between stations, share a single communications line (such as TELEX/TWX datacom lines), and have access to larger, expensive computer systems. ETHERNET features a high-speed data communications protocol that provides an almost instantaneous transfer of data. The problem with ETHERNET is that in addition to its high speed, it is also very high cost. A controller card for a microcomputer may well cost you $4000. When you stop and realize that one of the main reasons for using a network is to save the expense of putting a $2,000 printer at every station you begin to realize that ETHERNET is not yet cost effective. Luckily, there are alternatives to ETHERNET.

The Shape of Things to Come: Network Topology

A networking system is used to connect several computer systems together. Each computer system within a network is called a node or a station. There are all kinds of ways to connect several nodes together, any child playing with tinkertoys can attest to that fact. In a networking system there are various layouts possible when connecting a group of computer systems together; the layout, or graph, of the computer network is called the network topology. While there are infinitely many ways to connect up computer systems in a network there are three basic topologies from which most others can be derived. These topologies are the STAR, RING, and BUS (or tree) topologies.

The STAR topology (see figure 2) consists of some central node through which all communication must pass. The STAR topology is probably the most obvious since this type of layout is used by large mainframes and minicomputers which are connected to computer terminals. In a networking system, if a node other than the central node wished to transmit data to any other, the transmission would have to go through the central node in order to reach the destination node. This form of communication has two drawbacks. First, if the central node goes down so does the whole network. Secondly, if the central node is slow, the entire network will be slowed down since all communication must go through the central station. Even if the central node is fast, attaching additional stations to the system will cause severe network degradation since the central node becomes vastly overworked. The STAR topology does have one highly redeeming value. Much of the necessary connecting hardware is in place in many business offices-the PBX telephone system. The PBX (or private branch exchange) is the little operators console you find on the desk of the front receptionist in any large corporation. While intended to handle voice transmissions, most PBX's are easily upgraded to handle data transmission. The PBX system is a good example of a network using the STAR topology. Since all the wires are already in place, many businesses will opt to use their existing equipment thus saving thousands of dollars in installation costs.

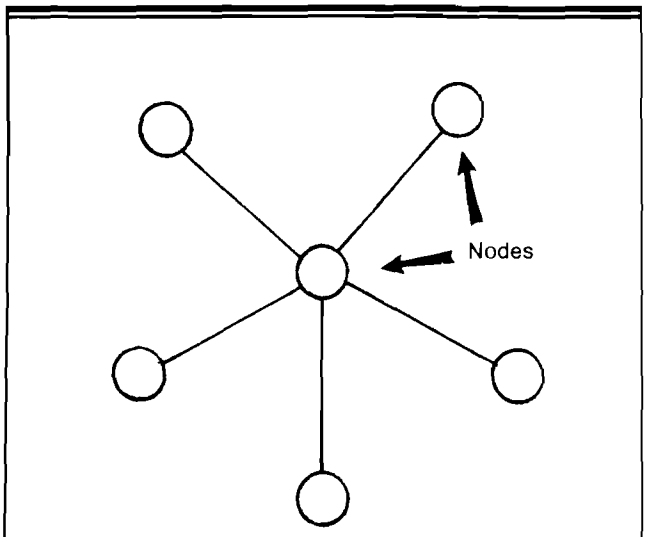


Figure 2: **Star Topology.**

The RING topology (see figure 3) was developed mainly for remote networks between large computer systems. The basic idea behind a ring network is that data packets

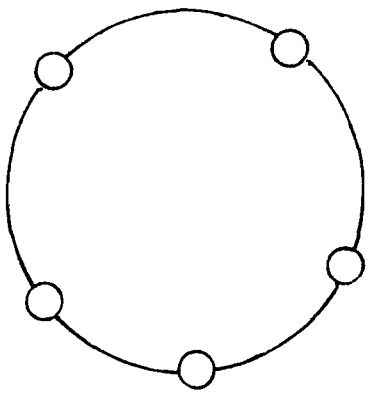


Figure 3: **Ring Topology.**

are passed from machine to machine in a "hot potato" fashion until the data packet arrives at the destination system. Since few local networks use the RING topology, there's little need to discuss it any further here.

The BUS topology (see figure 4) is the favorite topology amongst the various local networks. ETHERNET, OMNINET, NET/80, SOFTWORKS, Z/NET, and most other local networking systems use some variation on the BUS topology. The BUS topology is high robust. Any component (with the unlikely exception of the connecting cable) in the network system can fail and the network will still operate. The BUS topology allows you to add and remove nodes at will (in some cases without powering the system down). The BUS topology has another big advantage: often the circuitry required is less expensive than that required for other networking topologies.

Network topology is very important in remote networks where delays due to routing can be significant. Many remote networks use a store and forward approach where one station transmits data to an adjacent station from which point it is transmitted on down the line. Since it takes time to receive and retransmit a package of data, delays grow proportionally to the number of systems in a ring topology network (ring topologies are in many large networking systems). Since the network nodes are stationed fairly close to one another in a local network, the bus topology is normally used. The bus topology removes the problem of routing delays, but requires special cables and hardware which would be much too expensive for widely separated stations.

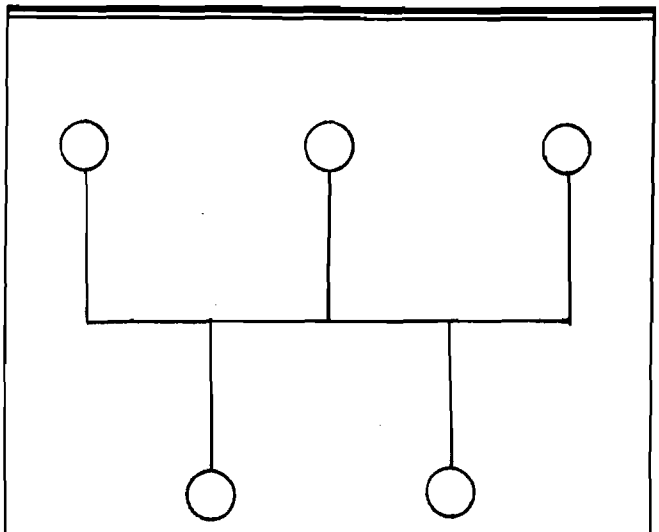


Figure 4: **Bus Topology.**

Data Transmission in a Networking System

Since the same transmission medium is used by all the nodes on a network, some supplemental information must accompany each data transmission. For example, a destination address must be tacked onto every transmission so that the intended receiver can recognize a data transmission. Other auxiliary information, such as a length field, type field, acknowledgement field, and checksum may be tacked onto the data.

A block of data containing the necessary source, destination, checksum, data, and other optional fields is called a frame or data packet. Data packets are constructed at each level in the network protocol. For example, the application layer sends data (possibly a character at a time) to the session layer where it is sent to the message/transport layer with a destination and source address. The message/transport layer collects the data until a complete frame is constructed, adds any required length, type, and other necessary fields and passes the data packet to the data link layer for transmission. The data link layer adds the checksum and any necessary acknowledgements and transmits the data over the network.

One important feature of a networking system is its data transmission rate. The data transmission rate is usually measure in bits per second (BPS) which is also called BAUD. For example, the familiar 300 BAUD line used in telecommunications is a three hundred bits per second data transmission rate. Many remote networks operate at 300 BPS and 1200 BPS (using leased lines). These data rates, however, are very slow. Most local networking systems operate in the range of one to twenty MBPS (megabits per second) which is 1,000 to 60,000 times faster than the lines used by the remote networks. The maximum data transmission rate of a networking system is called its bandwidth. Even though a network is capable of operating at 1 MBPS, the medium can operate at 300 BPS if need be.

In almost any networking system there isn't constant traffic on the net. So although a cable (or whatever) is capable of handling data at high speeds, the average data transmission rate is usually much lower than the bandwidth of the transmission medium. If the average bandwidth is low because no one is transmitting data, it's no big deal. After all, the average network user isn't interested in the quality of the cable being used to connect nodes, he's only interested in response time. If the average data transmission rate is only 50 BPS on any given day, but users get instantaneous response then things are just fine. But if the network is noisy, congested, or has some other problems, many packets will have to be retransmitted thus cutting the average transmission rate and the response time down considerably.

At any given time only one data packet can be in the process of being transmitted over a single circuit. If two nodes attempt to transmit over the same wire at the same time both transmissions will be garbled up and the transmission will be completely invalid. Such an occurrence is called a collision. Collisions occur most often in a network using a bus topology since must nodes in such a system share the same transmission circuitry. Receiving stations have no way of determining if a collision has occurred. If fact, the only time a receiving station can determine that a collision has occurred is when it performs some sort of error check and the checksum doesn't match.

One of the major problems with a collision is that the entire data packet must be retransmitted. Which means that the average data transmission rate is reduced. This is compounded by the fact that data transmissions occur asynchronously. That is to say anybody can start transmitting whenever they please. So halfway through your transmission someone else could start talking and wipe out your entire message. This problem is easily solved by by first listening to the bus to see if anybody else is transmitting. If someone else is using the transmission medium you wait until they are through. If there is no traf-

Products for Commodore, Atari, Apple, and others!

NEW THE MONKEY WRENCH II
A PROGRAMMERS AID FOR ATARI 800
NEW AND IMPROVED — 18 COMMANDS
PLUGS INTO RIGHT CARTRIDGE SLOT

If you are a person who likes to monkey around with the ATARI 800, then THE MONKEY WRENCH II is for you!! Make your programming tasks easier, less time-consuming and more fun. Why spend extra hours working on a BASIC program when the MONKEY WRENCH can do it for you in seconds. It can also make backup copies of boot type cassette programs. Plugs into the right slot and works with ATARI BASIC cartridge.

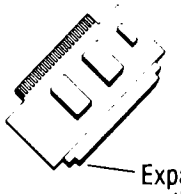
$59.95

The MONKEY WRENCH provides 18 direct mode commands. They are: AUTO LINE NUMBERING — Provides new line numbers when entering BASIC program lines. RENUMBER — Renumbers BASIC's line numbers including internal references. DELETE LINE NUMBERS — Removes a range BASIC line numbers. VARIABLES — Display all BASIC variables and their current value. Scrolling — Use the START & SELECT keys to display BASIC lines automatically. Scroll up or down BASIC program. FIND STRING — Find every occurrence of a string. XCHANGE STRING — Find every occurrence of a string and replace it with another string. MOVE LINES — Move lines from one part of program to another part of program. COPY LINES — Copy lines from one part of program to another part of program. FORMATTED LIST — Print BASIC program in special line format and automatic page numbering. DISK DIRECTORY — Display Disk Directory. CHANGE MARGINS — Provides the capability to easily change the screen margins. MEMORY TEST — Provides the capability to test RAM memory. CURSOR EXCHANGE — Allows usage of the cursor keys without holding down the CTRL key. UPPER CASE LOCK — Keeps the computer in the upper case character set. HEX CONVERSION — Converts a hexadecimal number to a decimal number. DECIMAL CONVERSION — Converts a decimal number to a hexadecimal number. MONITOR — Enter the machine language monitor.

In addition to the BASIC commands, the Monkey Wrench also contains a machine language monitor with 16 commands used to interact with the powerful features of the 6502 microprocessor.

NEW VIC RABBIT CARTRIDGE AND CBM 64 RABBIT CARTRIDGE

"High-Speed Cassette Load and Save!"



$39.95 (includes Cartridge and Manual)

Expansion Connector on the VIC Cartridge

"Don't waste your Life away waiting to LOAD and SAVE programs on Cassete Deck."

Load or Save 8K in approximately 30 seconds! Try it — your Un-Rabbitized VIC takes almost 3 minutes. It's not only Fast but VERY RELIABLE.

Almost as fast as VIC Disk Drive! Don't be foolish — Why buy the disk when you can get the VIC Rabbit for much, much less!

Easy to install — it just plugs in.
Expansion Connector on rear.
Works with or without Expansion Memory.
Works with VIC Cassette Deck.
12 Commands provide other neat features.

Also Available for 2001, 4001, and 8032

TELSTAR 64

Sophisticated Terminal Communications Cartridge for the 64.

PFO I0D O0D CP<D1>D2 BELL 12:30:00 10:14:36
(TELSTAR's Status Line)

Don't settle for less than the best!

- Upload/Download to/from disk or tape.
- Automatic File Translation.
- Communicates in Industry Standard ASCII.
- Real-Time Clock plus Alarm Clock.
- Line editing capability allows correcting and resending long command lines.
- 9 Quick Read functions.
- Menu-driven.
- Similar to our famous STCP Terminal package.
- Works with Commodore Modems and supports auto-dialing.

The best feature is the price — **only $49.95** (Cartridge and Manual)

Machine Language Monitor Cartridge

for the CBM 64

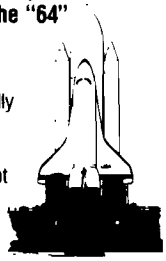
More than 20 commands allow you to access the CBM 64's Microprocessors Registers and Memory Contents. Commands include assemble, disassemble, registers, memory, transfer, compare, plus many more.

Someday every CBM 64 owner will need a monitor such as this.

Cartridge and Manual — **$24.95**

More than just an Assembler/Editor! Now for the "64"

It's a Professionally Designed Software Development System



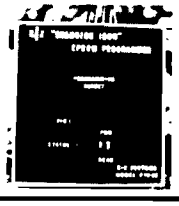
MAE
for
PET
APPLE
ATARI
~~$169.95~~
Only $59.95

NOW, The Best for Less!

- Designed to improve Programmer Productivity
- Similar syntax and commands — No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI.
- Coresident Assembler/Editor — No need to load the Editor then the Assembler then the Editor, etc.
- Also includes Word Processor, Relocating Loader and much more
- Join the ATUG User Group for MAE formatted disks
- STILL NOT CONVINCED? Send for free spec sheet!

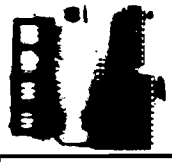
ATARI, PET, AND CBM 64 EPROM PROGRAMMER

Programs 2716 and 2532 EPROMs. Includes hardware and software. PET = $75.00 - ATARI and CBM 64 (both include sophisticated machine language monitor) = $119.95



CBM 64 Debugger

A more sophisticated Machine Language Monitor/Debugger. 20K of object code makes this a powerful tool. Works as a symbolic debugger for the MAE assembler. Diskette and Manual — **$49.95**

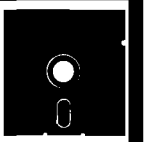


TRAP 65

TRAP 65 is a hardware device that plugs into your 6502's socket. Prevents execution of unimplemented opcodes and provides capability to extend the machines' instruction set.
For PET/APPLE/SYM
Reduced from $149.95 to $69.95

5¼ INCH SOFT SECTORED DISKETTES

Highest quality. We use them on our PETs, APPLEs, ATARIs, and other computers. $22.50/10 or $44.50/20



Prowriter Printer - Excellent dot matrix print Parallel = $489.00
Serial = $600.00 IEEE = $589.00

DC Hayes Smart Modem = $235.00
DC Hayes Micro Modem II = $289.00

Rana Disk Drive - 375
4 Drive Controller - 114

EPROMS 2716 = $4.50 2532 = $7.50
Over 40 Commodore Programs by Baker (on 4040) = $25.00

Eastern House

3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA MasterCard

fic on the network then you can start transmitting. This method, called the Carrier Sense Multiple Access (CSMA) improves efficiency tremendously. CSMA does not completely eliminate collisions, but it does cut them down tremendously.

A collision can still occur if two stations happen to look at the bus at the same time and find that the carrier is absent. Since data transmission takes place at somewhat less than the speed of light there is a finite (although small) time delay between the time a station puts a carrier signal on the network and it travels to all the stations. This time, although measure in microseconds, is quite long compared to the cycle time of a typical personal computer. For example, in a normal local networking system this time interval, called the round-trip transmission time might be 30-60 microseconds. If two nodes attempt to transmit data within one round trip transmission time there will probably be a collision. Collisions in a CSMA collision detection system are quite rare unless there is an enormous amount of traffic on the network.

An Introduction to the NESTAR CLUSTER ONE

There are hundreds of different networking arrangements, with new ones being introduced everyday. Therefore, in order to discuss some specific details a specific system must be discussed. The remainder of this paper will concern itself with the NESTAR CLUSTER/ONE MODEL A local networking system.

The Cluster/one lets you connect up to 64 Apple II computers together in a local network. Each Apple is interfaced to the system via a plug-in card that attaches to the Apple's bus. This circuit card contains the necessary interface circuitry as well as 1K of RAM and 2K of ROM. Nodes in the Cluster/one network are interconnected with 16-conductor cable similar to that used to attach disk drives to an Apple II.

The Cluster/one transfers data at 240 KBPS, quite slow for a local network. Slow, however, is a relative term. 240 KBPS is roughly 1,000 times faster than the 300 BAUD lines used by the SOURCE and other telecommunications systems. 240 KBPS is the maximum data rate sustainable by software methods using a 1 Mhz 6502. By comparison Xerox's ETHERNET transfers data at 10 MBPS yet the circuitry required to support ETHERNET is better than ten times the cost of the Cluster/one. In general, the 240 KBPS is fast enough. To put it into concrete terms, you can transfer 64K of data in less than four seconds using a 240 KBPS data transfer rate, that's much faster than a floppy disk or printer.

The Structure of the NESTAR System

The NESTAR Cluster/one implements three of the lowest four levels defined by the ISO reference model of Open Systems Interconnection (OSI). These are the physical, data-link, and message/transport layers. The networking layer is usually not present in a local network, and is absent from the Cluster/one networking system. The remaining layers (session, presentation, and application) are either supplied by the user or left unimplemented. How the lower layers are implemented is determined by the configuration of the network and the software purchased with the system.

One of the most important network design considerations is the network topology. The Cluster/one normally uses the bus topology. Due to the structure of the hardware almost any topology can be accommodated. Typically, however, the bus topology is used because it is the least expensive and the software was designed with the bus topology in mind.

A typical NESTAR consists of several user stations (clients) and one or more file server stations. A file server consists of a dedicated Apple computer with an 8" floppy or 14" hard disk. All the other users on the net share the resources provided by the file server. The disk drives supported by the Cluster/one provide 1.2 megabytes to four Gigabytes of disk storage using one or more drives on the system. In addition to the file server, the dedicated Apple also provides a clock/calendar server so that files sent to disk can be "time-stamped" with the creation and update time. The Cluster/one network interface cards were designed using low-cost non-critical components in an effort to keep the total package cost down. It would be insane to require a $4,000 interface card on a computer system that only costs $2,000 (don't laugh, Intel is selling ETHERNET interface cards for $4,000). The network interface was optimized for low-cost and reliability at the expense of speed. As mentioned previously, the bandwidth of the NESTAR system is quite low.

The Cluster/one interface card uses a passive interface. This gives you the ability to power up and power down stations on the net without adversely affecting the network. It also allows you to add new nodes to the system without having to power the system down or even halt network communications. In the event of a station failure, the network may continue to function providing the controller card itself isn't damaged. In many topologies (like the ring and star topologies) the failure of one or more stations can shut the whole network down. In the case of the Cluster/one, the net will continue to function as long as at least two stations are functional.

The NESTAR system using a 16-conductor cable to connect nodes in the net. This is the second unusual feature to the Cluster/one system, most local networks use a twisted pair or coax cable (two conductors). Of the 16-conductors, eight are used to transmit data in a byte-serial fashion. By transmitting the data eight bits at a time the effective bandwidth is increased by eight. Although the system is transferring data at around 240 KBPS, the data is actually being transferred at a data rate of 30 KBPS on any given data wire. This gives the 6502 roughly 34 cycles in order to perform the necessary checksum calculations and store the data in memory or fetch the next byte from memory. An additional line is used for the carrier signal. When active, someone is attempting to use the network, when inactive the network is free and ready for transmission. Three of the remaining seven lines are used for handshake using the standard data available/data taken/ready to accept handshake protocols. The remaining four lines are signal grounds.

To transmit data over the network a controller card first checks to make sure everyone is listening and then it places the data on the bus. The data available line is set active and the destination node senses the data available line and reads the data off of the bus. Once the data is read the data taken signal is set active and the source station may transmit additional data. Due to the structure of the net-

work, it's real easy to reverse the direction of the data lines and acknowledge a data transmission. This fact is used by the NESTAR software to handle frame acknowledgements as will soon be pointed out.

The Cluster/one system is highly tolerable of transmission errors. First of all its low bandwidth allows inexpensive flat ribbon cable to be used instead of expensive coaxial cable. Secondly, rather than use extra-cost passive terminators in the system, the software was designed to accept unterminated connections by providing a software delay whenever data was placed on the bus. This delay allowed all the ringing and echoes to die down before attempting to read the data on the bus. By running at such a slow data rate (30 KBPS/conductor) many of the problems associated with transmission lines is avoided.

All data transmitted over the NESTAR CLUSTERBUS (a possibly trademarked name for the 16-conductor interconnect) is transmitted in a purely digital form. No phase encoding (like that used by ETHERNET) is performed, nor are the signals converted to some type of analog signal before transmission. This simplifies the data conversion process and likewise lowers the cost of the NESTAR system.

The NESTAR Data-Link Layer

The data-link layer in the Cluster/one is implemented in hardware and firmware. The software required for the data-link layer is contained in the 2K byte ROM on the NESTAR interface card. The data-link layer is responsible for transmitting data onto the net and accepting data from the net. The data link layer attaches a checksum to the end of a transmission and checks the checksum value at the end of a transmission. It is responsible for handling retransmissions in the event the data wasn't transmitted properly. The data link layer is also responsible for media-access scheduling, addressing, flow control, and congestion control. To sum it up, the data link layer is responsible for providing error-free data transmissions between two stations on the network.

Transmitting data:

The data link layer is presented with a data packet from the transport layer. The data packet is of variable length and consists of 4n bytes where n is the number of bytes of data to transmit. The first byte transmitted is the source address. This is used to detect a collision on the Clusterbus. If two stations attempt to place their source addresses on the bus at the same time at least one of the transmitting stations will recognize a collision since all source addresses are different. If a collision does occur, all stations recognizing the collision will jam the bus to make sure that everyone else sees the collision as well. The data-link layer only checks for collisions during the first few bytes of transmission. Later collisions during the transmission of the data frame are highly unlikely since the carrier signal is checked before attempting to transmit data. In the case some station erroneously transmits data a checksum error will occur and both packets will have to be retransmitted. By not checking for collisions after each transmitted byte, the data transmission rate is increased.

After the cable has been allocated (by placing the source address on it) the source station places the destination address on the bus and, after a suitable delay, activates the carrier control line. On the active edge of the carrier control line hardware on each of the interface cards triggers an address comparator that compares the address on the Clusterbus to the source address of the particular card. If a match is made then the software in the destination interface will be activated in order to read the data packet being transmitted.

After the destination address is placed on the bus and accepted by the source station, the type and length bytes are transmitted. Following the length byte, "length" bytes of data are transmitted over the network. During the transmission of the type, length, and data bytes the source station is constantly computing a checksum. At the end of the data transmission the 16-bit (two byte) checksum is transmitted over the network.

Once the checksum is transmitted the data packet transmission is complete-almost. The only thing remaining is to make sure that the data packet was properly received. In order to check that the data packet was correctly sent the source station reverses its data lines and becomes a receiver. The destination station transmits a single byte containing a ACK (acknowledgement) or NAK (negative acknowledgement) code. If a NAK code is received then the source station retransmits the data frame beginning with the type field. If an ACK code is received then the source station frees the network bus by setting the carrier line inactive.

Data Reception:

Receiving data is almost the converse operation to transmission. Data reception begins when the destination hardware recognizes its address on the bus. Once the destination hardware accepts the destination address from the bus it reads the type and length fields respectively. The destination interface software begins calculating the checksum with the receipt of the type byte. Once the length byte is accepted the destination station reads that many data bytes from the bus. After the data bytes are read two checksum bytes are read and the final checksum is checked. If the checksum is ok then the destination station transmits an ACK code to the source station. After the ACK is transmitted the destination station goes on about its business. If the checksum didn't match, then a NAK code is transmitted to the source station and the destination station waits for a retransmission.

Congestion and flow control are also taken care of by the data-link layer. Flow control (making sure the source doesn't transmit data any faster than the destination can receive it) is handled automatically by the hardware. A new byte isn't transmitted until the old one is taken. Flow control isn't quite that simple however. For example, if a destination station is powered down in the middle of a transmission the source station would be left hanging in a locked-up state if a timeout mechanism was not provided for. Checking for timeouts also comes under the heading of flow control since making sure that the destination station is accepting the data fast enough is just as important as making sure the transmitting station doesn't transmit data too fast. This prevents stations from monopolizing the network if for some reason they decide to start accepting data at only one byte per minute.

Congestion control is possibly more important than flow control in the Cluster/one. Congestion control deals

with preventing too many users from accessing the network at any given time. Congestion control is handled in one of several methods. Since the Cluster/one uses a CSMA collision detection the number of collisions is reduced tremendously. Whenever a large number of stations start wanting to use the bus, however, the number of collisions rises dramatically. While the bus is jammed and garbage is being transmitted the average data transfer rate is sorely reduced. Worse than that, it's reduced at the time it's needed the most when there are a lot of stations requesting network service.

When a collision is detected, the stations recognizing the collision back off of the network (i.e., attempt to set the carrier line inactive) and jam the bus so that all stations on the Clusterbus recognize the collision. Once the jamming is complete each station attempting to transmit waits a small, random amount of time and tries again. It should be obvious why a random amount of time is used, if everyone waited the same amount of time a collision would be sure to re-occur. By waiting a random amount of time (instead of a fixed, but different, amount of time) all stations will be given equal priority on the bus. Another responsibility of the congestion control is to make sure that a pair of stations don't hog the bus. As you will soon see it is possible to hold the carrier line active while several data frames are transmitted. The congestion control portion of the data-link layer is responsible for making sure no one hogs the bus for too long.

The Message/Transport Layer

The next layer of protocol in the Cluster/one system is the message/transport layer which manages the handling of messages and conversations among stations. Due to the fact that the length of a data packet is limited to 260 bytes (256 bytes of data, two bytes for type and length, and a two-byte checksum) data must be transmitted in a datagram fashion. Remember, in a virtual circuit operation the source and destination stations connect just once and then all data is transmitted across the line until the transmission is complete. In a datagram system data packets, or datagrams, are transmitted a block at a time with the receiving station making heads or tails of the received data. Since you can transmit a maximum of 256 bytes in a data frame on the Cluster/one the system is limited to datagram service. Since, in the Cluster/one, datagrams are always transmitted in order and are never lost, a sort of "pseudo-virtual circuit" operation can be achieved by transmitted datagrams back and forth between two stations.

The simplest form of communication is a one-way, single-packet transmission (the datagram). Datagrams in the NESTAR system are always acknowledged (as discussed in the section on the data-link layer). One-way communications, however, are only moderately useful. A truly useful communication system requires a series of two-way communications. This type of virtual circuit operation is easily accomplished by sending datagrams back and forth between the two stations. Such action, of course, requires a certain amount of cooperation between the two stations.

One of the more interesting features of the Cluster/one system revolves around the carrier line. By leaving this line active inbetween packet transmissions a multiple packet transmission can be accomplished without any interruption on the bus. While this does provide a small increase in performance (especially when the packets being transmitted are small), What's more important is that the data frames can be exchanged without interruption. This very important operation allows the construction of indivisible operations like the "set and test" instructions required by operating systems. Thus, semaphores can be efficiently implemented in the NESTAR network. Care must be taken, however, to make sure that two processes don't accidentally hog the bus. As was previously mentioned, the data-link software checks certain time-out counters to make sure that the current process isn't hogging the Clusterbus.

Network Transparency

One of the most important features of a hardware system like a network is that it must be totally transparent to the user. A networking system would be absolutely useless if you had to throw away all the software you've written and start over again to satisfy the network requirements. Such a horrible thought need not come true, due to the structure of the network itself, it is perfectly possible to treat the network as though all the devices attached to the network were, in fact, attached to the individual work station. This concept of virtual I/O makes the system completely transparent to the end user. He need not change the way he thinks about the file system or I/O. Almost all of the existing software will run unmodified on the network. And the modifications required to adapt existing single-user applications for a multiple-user environment are easy to make.

Some of the stations on the network (the servers) are designed to serve other stations (the clients). The printer server is a good example. This consists of an Apple II attached to a printer connected to the network. From the user's point of view he need only type PR1 (from BASIC) or OPEN{F,"PRINTER:"}; (from Pascal) in order to access the printer. Everything else is handled by the network and patches made to the resident operating system. The only difference is that the printout comes from the printer server instead of the peripheral connected to slot one in the users Apple.

To the Apple user the most important feature the Cluster/one offers is that it forces very little change on current work habits. The system is easily learned and the software enhances productivity through its combined computational and communication facilities. In any corporate structure communication is essential. A group of widely separated computer systems doesn't enhance communication throughout the corporation. Unfortunately, although a timeshare system provides the communications capabilities response time is not very good. The Cluster/one network allows several microcomputer system to communicate with one another while maintaining the excellent response time microcomputer owners are used to and without obsoleting existing software.

MICRO

Randall Hyde is vice president in charge of advanced research and development at Lazer MicroSystems, Inc., a Southern California software development firm. You may contact Mr. Hyde c/o Lazer MicroSystems, Inc., 1791 Capital, Unit G. Corona, CA 91720.

Downloading BASIC on the PET • VIC-20 • *C64*

by
Kevin Stone
and
Andrew Cornwall

Tired of typing in a listing from a friend or having to travel to a friend's place to copy a program you would like? If so, continue reading. This article explains a simple method to send and receive BASIC programs (or machine language programs with a BASIC loader) over the telephone using a VIC, C-64 or PET. Equipment requirements are minimal, consisting of the microcomputer and compatible modems connected to the serial port of the VIC or C64 or the IEEE port of the PET. The program transferred is sent as a listing and automatically converted by the dynamic keyboard technique to tokenized (program) form by the receiving computer. After communication, the received program can be readily run, or saved to cassette or disk. Programs can be received from any other system.

Kevin Stone, VIC enthusiast, is a former Equipment Technician for CNCP Relecommunications, and is currently a Technical Support Representative for Victor (Canada) Ltd. Kevin resides in Halifax, Nova Scotia, Canada.

Andrew Cornwall, is a PET hobbiest with a wide range of interests in microcomputer applications. He is Secretary of the Nova Scotia Commodore Computer Users Group. He resides in Dartmouth, Nova Scotia, Canada.

Sending a Program

The first step in transferring a program is to coordinate the modems at the sending and receiving ends. As usual for any communication, one modem should be in the originate and the other in the answer mode (for the Bell 103 standard modem). Although it does not seem critical, we prefer setting both modems at full duplex, if there is a duplex option.

To send a program, load it into the transmitting computer's memory in the usual way (either from tape or disk, or direct entry). Once telephone contact has been established between modems, the transmitting computer lists the program to its modem device. The procedure to do this is slightly different for the VIC or C64 and PET, and each is described separately below. Note that if only a portion of a program is to be transferred, the LIST command would contain the appropriate line number references (e.g. LIST 50-, LIST 50-100, etc.).

VIC or C64 Transmit

In direct mode, type the following lines and hit return after each. Serial port parameters are set by **CHR$(6)** at 300 baud, no parity with 8 data bits, full duplex, 1 stop bit, and three line handshake. This will work with most modems, but some may require slightly different parameters.

```
OPEN 2,2,0,CHR$(6)
CMD 2
LIST    (with line range if
         required)
```

The program will now be sent over the modem as a full listing - not in tokenized form. When listing is finished, indicated by return of the screen cursor, wait about 15 seconds (or if the modem has a transmit light, until this light stops flickering) for the serial port buffer to empty to the modem, then type in:

```
PRINT2
CLOSE 2
```

This redirects normal output to the screen and closes the file.

PET Transmit

Enter the following lines in direct mode and hit return after each. The open statement assumes that the modem is device number 5, a common but not universal modem device number. Use the correct number for your system.

```
OPEN 5,5
CMD 5
LIST
```

When listing is finished and the screen cursor returns, enter:

```
PRINT5
CLOSE 5
```

Transmission Time

The time needed to send a program is governed by the modem baud rate. At 300 baud, transmission requires about 30 seconds per kilobyte of listing. Since a program listing can be as much as twice as "long" as its tokenized program form, a 4K program in memory can take up to four minutes to transmit as a listing.

Receiving a Program

The receive method uses the dynamic keyboard append technique. While the routines for the VIC, C64 and PET are somewhat different, the general idea is the same. In each case a program listing is taken in and appended line-by-line to the small receive program. When the process is completed, the total program in memory consists of the receive routine plus the communicated program. The receive routine can be removed by using a toolkit delete command or simply by entering the line

numbers for the routine and hitting return after each.

As with any append process, the line numbers of the program being transmitted should not overlap those in the receive routine. If the line numbers do conflict, then simply renumber the program. The **GOTO**'s must be fixed along with the renumbering. So long as overlapping line numbers is not a problem, we prefer using the receive routine with low line numbers, since it is technically faster (and easier to delete!).

Upon being run, the receive routine will ignore all transmitted characters (such as the word **READY** sent after the **CMD** command) until a numeric character with a value 1 to 9 is encountered. The routine presumes that the first numeric character is part of the initial line number of a listing, and commences to take in the remainder of the listing after a valid number is received.

At some point in the routines dynamic keyboarding takes over to carry-out the listing-to-program append process. This is visible as each communicated program line appears on top of the screen, a transfer line (ending with a **GOTO** statement) is printed under it, next **READY** flashes once, the screen clears and the sequence starts over. This will continue until the word **READY**, which is automatically sent at the end of the listing, is encountered. At this point the process will stop with a harmless **OUT OF DATA ERROR**.

VIC and C64 RECEIVE Programs

These routines take advantage of the serial port 256 byte input buffer, which makes the timing of the receive and append processes less critical, although on average the routine keeps up with the character input rate at 300 baud. Upon being run the routine clears the screen and waits for the listing to be transmitted. When listing starts the action of dynamic keyboarding can be seen on the screen. The routine simultaneously receives and appends the incoming program listing. The buffer prevents data being lost during the dynamic keyboarding phase of the receive/append process.

Although the input buffer takes only 256 bytes, the actual serial port buffer occupies 512 bytes. The additional 256 bytes is for a transmit buffer, which is not used when receiving. After allowing for the buffers and receive routine itself, only about 2K of memory is available on a basic VIC for the received program. The routine will work with any memory expansion configuration however, without modification to the routine.

Receive Program Detail

LINE 0: **OPEN**s a file to the serial port. **CHR$(6)** sets port parameters to 300 baud, no parity with 8 bits, full duplex, 1 stop bit, and 3 line handshake. These parameters are appropriate for most common modems. The screen is then cleared.

LINE 1: **GET**s a character via the modem and tests for receipt of first numeric character 1 through 9.

LINE 2: **PRINT**s first received numeric character to screen.

LINE 3: **GET**s via the modem and **PRINT**s across the screen. End of program line is determined by receipt of a carriage return.

LINE 4: **PRINT**s dynamic keyboard transfer line to the screen. **POKE 152,1** tells the VIC/C64 that one file remains open; **GOTO 6** instructs the routine to restart at **LINE 6** when the transfer line is executed in simulated direct mode.

LINE 5: Fills keyboard buffer locations **631** to **634 with a home cursor** and three **carriage returns**; **POKE 198,4** tells the VIC/C64 that the keyboard buffer contains four characters. After this the routine is formally instructed to **END**. In fact, however, the routine continues as the characters in the keyboard buffer are executed (simulating direct mode operation) which enters (appends) the transmitted program line printed to the screen and activates the transfer line.

LINE 6: Clears the screen in preparation for the next received line. It then executes a **GOTO 3** to continue the input.

Using Other Line Numbers

Since many programmers use the convention of starting at **LINE 10**, the above routine which resides from **LINE 0** to **LINE 6** should not normally be in the way of the program being transferred. If there is a conflict, then simply renumber the BASIC program, remembering to change the line numbers for the three **GOTO** commands in lines 1, 4 and 6.

PET Receive Program

The PET IEEE port does not have an input buffer. As a result, when simultaneous receive and conversion is attempted, incoming characters are missed during dynamic keyboarding. To overcome this problem, the receive and conversion operations are done separately. First, the listing is received and **POKE**'d verbatum into a protected area of memory above 2K. When receiving is finished, sensed by a 64 millisecond time-out error, the listing is automatically **PEEK**'d from memory and dynamic keyboard appending commences.

Overall, the PET routine is slower than the VIC/C64 version, since it takes about an extra second per line to perform the separate appending process. Another disadvantage is that initial storage of the full listing requires as much as twice the memory as its counterpart in tokenized program form. Very large programs, however, can be communicated in stages, saved, and appended together afterwards.

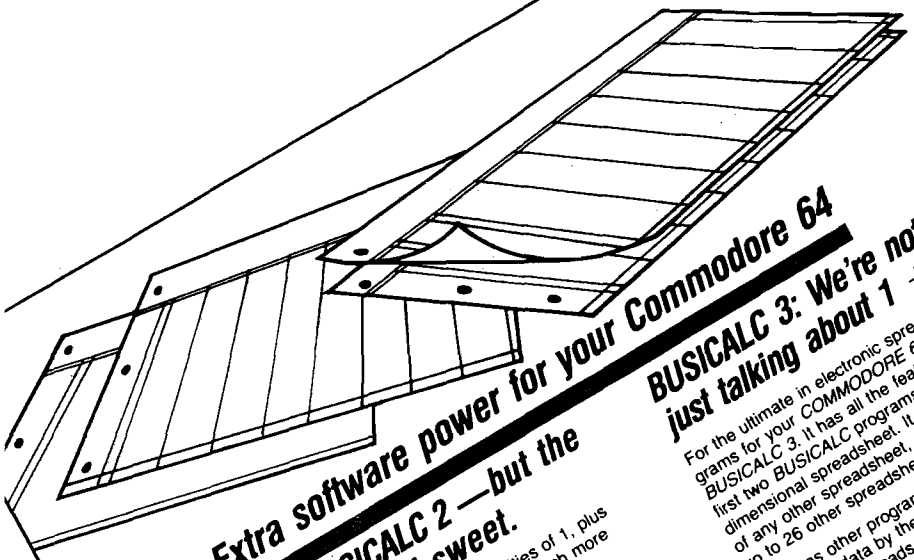
When run, the PET routine clears the screen and prints the message "**READY TO RECEIVE**". When a listing is being received, the message "**RECEIVING**" is printed. When the receiving is completed, i.e. a **READY** is received, the dynamic keyboarding takes over.

PET Receive Program Details

LINE 0: **POKE 53,8** sets top of BASIC memory at 2K to protect higher memory for character storage. It then **OPEN**s a file for device number **5**, which is presumed to be the modem

```
VIC

 0 OPEN 2,2,0,CHR$(6):PRINT "□";
 1 GET A$:IF VAL(A$)=0 THEN GOTO 1
 2 PRINT A$;
 3 GET #2,A$:PRINT A$;:
   IF A$<>CHR$(13) THEN 3
 4 PRINT:PRINT "POKE 152,1:GOTO 6"
 5 POKE 631,19:POKE 632,13:POKE 633,13:
   POKE 634,13:POKE 198,4:END
 6 PRINT "□";:GOTO 3
```



Extra software power for your Commodore 64 with BUSICALC 2—but the price is still sweet.

BUSICALC 2 has all the capabilities of 1, plus features that you can only find in much more expensive programs. It has a total of 17 commands, for example, including:

- "Save with Replace," a popular feature that lets you easily keep track of data and make revisions;
- Formula print-out;
- "Window framing," a great command that retains row and column headings even as you scroll through a long or wide worksheet, thus reminding you of what the numbers represent;
- A high-speed "find" capacity for creating parts lists and inventories;
- Direct disk commands without leaving your spreadsheet;
- Overall setting of the number of decimal places with exceptions allowed anywhere;
- User defined functions;
- Direct erase command for all or any specified part of your spreadsheet;
- Instant color change without leaving program;
- Direct quit function.

BUSICALC 2 adds 17 (count them, 17) additional built-in math and logic functions. They are Sine, Cosine, Tangent, Logarithm, Square Root, Round-off, Truncation, Absolute, Random, Standard Deviation, Count, and three Logic functions (Positive, Negative and Zero).

USER DEFINED FUNCTIONS	
KEYSTROKE	DEFINITION
W	V(1)+V(2)+V(3)
X	V(1)/(1+(V(2)/100)↑V(3))
Y	S(V(2)/2)+V(1)+(P(V(3)-V(4)))

BUSICALC 2 allows up to 999 rows and up to 125 columns. Your total spreadsheet size can be up to 2500 cells (boxes), far surpassing anything previously offered. Like its predecessor, *BUSICALC 2* comes with a 56 page user-friendly manual that has step-by-step procedures and a quick reference section. It also has a nice low price!

BUSICALC 3: We're not just talking about 1 + 2.

For the ultimate in electronic spreadsheet programs for your *COMMODORE 64*, look to the first two *BUSICALC* programs. It has all the features of the dimensional spreadsheet. It can retrieve data off of any other spreadsheet, and you can access up to 26 other spreadsheets at one time.

Whereas other programs are limited in their access to data by the memory in the computer, *BUSICALC 3* reads data off the disk drive. With *BUSICALC 3*, you can reach up to 52,000 boxes (or "cells") of information. This program provides all the power and flexibility a user could ever want in an electronic spreadsheet program.

Chartpak-64: A new companion program to BUSICALC.

This great aid can take data off your spreadsheet and give you back a bar graph or pie chart of that data. These displays look great in formal presentations and reports, but they can also make the implications of data clear. Chartmaster complements *BUSICALC* beautifully; it's a handy program to own.

BUSICALC 1 COMMODORE 64, disk or tape	**$49.00***
BUSICALC 2 COMMODORE 64, disk or tape	**79.00***
BUSICALC 3 COMMODORE 64, disk or tape	**129.00***
CHARTPAK-64 COMMODORE 64, only	**42.95***
BUSICALC 20 VIC-20, disk or tape	**49.95***

*There is an additional $3.50 US and Canada, $10.00 Europe and Asia, shipping charge per order. California residents add sales tax.

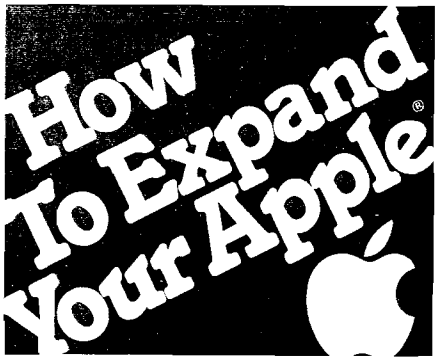
For CBM-64 and VIC owners only:

This is just 1 of 20 pages of the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full 7¼ x 9 splendor, and another 19 pages of peripherals, software and books that will make your CBM-64 or VIC computer even nicer to live with. So, if we missed sending you your very own copy within the last few weeks, call us at **(800) 227-9998 (unless you live in California, in which case call (415) 965-1735.**

From Skyles Electric Works, the oldest and largest professional specialists in the business.

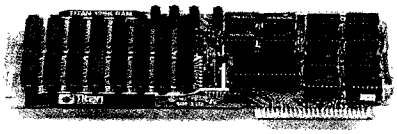
Skyles Electric Works
231-E S. Whisman Road
Mountain View, California 94041



More speed, more memory workspace from the people who lifted the 64K limit.

In 1981, we introduced our famous Saturn RAM card to boost the Apple's memory by 32K. Now, there are 64K and 128K versions for added power.

Increased RAM lets you run bigger programs, and our VisiCalc® expansion software provides a dramatic expansion of workspace memory Also, the extra memory can work as a pseudo disk for instant access. (No more 20-second searches!) The Saturn RAM board is a super aid for advanced word processing, data base management, spread sheet, and accounting applications. And perfect with PASCAL, CP/M®, and BASIC.



Use our easily installed cards in combination—and get a whopping 220K of VisiCalc workspace on the Apple IIe. Or sensational enhancements on Apple II, II+, Franklin, Basis and most Apple compatibles. We even have software that increases your Apple's memory in BASIC programming up to 4 megabytes!

This kind of performance requires sophisticated bank switching and supporting software that we originated, proved, and improved. Buy from the leader!

Let us help you expand your Apple's productivity. For information on our RAM boards and other Titan microcomputer products, see your computer dealer or contact: Titan Technologies, Inc., P.O. Box 8050, Ann Arbor, MI 48107; Telephone (313) 973-8422.

Sales and Marketing by The MARKETING RESOURCE GROUP, Costa Mesa, CA.

Titan
TECHNOLOGIES, INC.
FORMERLY SATURN SYSTEMS OF MICHIGAN

Apple is a registered trademark of Apple Computer, Inc.
VisiCalc is a registered trademark of VisiCorp, Inc.
CP/M is a registered trademark of Digital Research, Inc.

PET

```
 0 POKE 53,8:OPEN 5,5:J=2048:
   PRINT "■READY TO RECEIVE":GOTO 3
 1 GET#5,A$:IF ST<>0 THEN GOTO 5
 2 J=J+1:POKE J,ASC(A$):GOTO 1
 3 GET#5,A$:
   IF ST<>0 OR VAL(A$)
   =0 THEN GOTO 3
 4 PRINT "RECEIVING":GOTO 2
 5 POKE 53,128:K=J:J=2048:CLOSE 5
 6 PRINT "■";
 7 J=J+1:B%=PEEK(J):PRINT CHR$(B%);:
   IF B%<>13 AND K>J THEN GOTO 7
 8 PRINT:PRINT "K="K":J="J":GOTO 6"
 9 POKE 623,19:POKE 624,13:POKE 625,13:
   POKE 626,13:POKE 158,4:END
```

device number. Variable **J** is initialized to the 2K boundary. The screen is cleared and the message "**READY TO RECEIVE**" is printed.

LINE 1: **GET**s a character from the modem and tests the status variable **ST** for an error. If an error occurs, it is assumed to be a time-out error (**ST = 2**), and branches to the dynamic keyboard section of the program, starting at **LINE 5**.

LINE 2: Increments variable **J** and **POKE**s the PET-ASCII value of the latest character received into memory location **J**. Branches back to **LINE 1** to get another character.

LINE 3: Tests for receipt of first numeric character to suppress **READY** and other non-BASIC lines.

LINE 4: **PRINT**s message "**RECEIVING**", after the first numeric character is received.

LINE 5: Restores the normal top of BASIC memory. Defines variable **K** as last value of **J**, and redefines **J** as 2K. **CLOSE**s file number **5**.

LINE 6: Clear the screen.

LINE 7: **PEEK**s listing from memory and **PRINT**s characters across the screen. End of program line determined by carriage return (**= 13**).

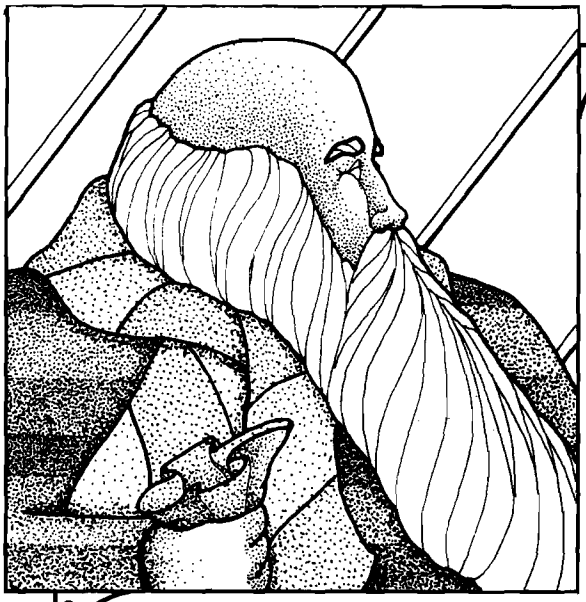
LINE 8 **PRINT**s dynamic keyboard transfer line to the screen. The transfer line carries over the values of **K** and **J** and set the program to restart at **LINE 6**.

LINE 9: Fills keyboard buffer locations 623 to 626 with a 'home cursor' and three 'carriage returns'; the **POKE 158,4** sets a variable to tell the PET that the keyboard buffer contains four characters. After this, the routine is formally instructed to **END**. In fact, the routine continues as the characters in the keyboard buffer are executed, simulating direct mode operation to enter (append) the received program line printed to the screen and to activate the transfer line.

Reliability

Even though this method for transferring programs does not use any error detection, such as parity checking or checksum, we found it to be very reliable. Phone line glitches can cause difficulties, but these were rare during our development and testing of this technique. On one occasion someone dialed an extension telephone while a program was being transmitted. On another, the phone line mysteriously went dead for several seconds. Normal phone line background noise and cross-talk did not seem to cause any problems. In our "formal" testing of this procedure, an 8K listing was communicated ten times over the period of several days. Each time the program was transferred successfully. Although our experience was on a local call basis, long distance telephone program communication should be feasible.

If a problems occurs when a VIC or C64 is receiving a program. the routine can be rerun to accept that portion of the program listing which was not received correctly. If a problem occurs while a PET is receiving, the correction procedure is more difficult. That portion of the program received in good condition can be saved and the routine rerun to take in the listing of the remainder. Afterwards, both parts can be appended together. It is probably more convenient to simply re-do the entire procedure from the beginning.



VICTORY SOFTWARE

WOULD LIKE TO WISH OUR CUSTOMERS

H • A • P • P • Y H O L I D A Y S

AND THANK THEM FOR THEIR PATRONAGE THROUGHOUT THE YEAR.

FEATURING PROGRAMS FOR THE VIC-20 AND THE COMMODORE 64.

BOUNTY HUNTER $24.95*

An adventure in the Old West. Journey back with us into the days of Jessie James and Billy the Kid where the only form of justice was a loaded revolver and a hangman's noose. In this full-length text adventure, you play the role of Bounty Hunter, battling against ruthless outlaws, hostile Indians, wild animals and the elements of the wilderness with only your wits and your six gun. Average solving time: 20-30 hours. If you love adventures, this one is a real treat.
Available for **COMMODORE 64** and the **VIC-20** (with 8K or 16K expander). Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

KONGO KONG $24.95*

Climb ladders, avoid the barrels the crazy ape is rolling at you, and rescue the damsel. Commodore 64 version features 4 different screens!
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

GRAVE ROBBERS $19.95*

Introducing the first GRAPHIC ADVENTURE ever available for the VIC-20 or COMMODORE 64! With realistic audio-visual effects, you explore an old deserted graveyard and actually see the perils that lie beyond.
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **KEYBOARD**.

CHOMPER MAN $24.95*

Don't let the bullies catch you as you gobble the goodies! This program has 8 screens and still fits in the standard memory.
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **JOYSTICK** or **KEYBOARD**.

THE • EARTH • WARRIOR • SERIES

METAMORPHOSIS $24.95*

You stumbled into the nest of the Cyglorx and find yourself fighting off robot tanks guarding the Cyglorx eggs. You think you have everything under control and then the eggs start hatching.
Available for **COMMODORE 64** and **VIC-20**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**

CREATOR'S REVENGE $24.95*

The creator assembled a massive army of robots and insects to take revenge on the earth. Destroy insects, get treasures, and get the neutron bomb deactivator. Battle robots and destroy the neutron bomb before it annihilates your city. Miss and you must face the mutants. Features 4 different screens.
Available for **COMMODORE 64**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

LABYRINTH OF THE CREATOR $24.95*

Journey into the most complex and dangerous fortress ever built by the creator. You will encounter deadly robots, skulls, lakes, avalanches, false creators, and a creature who roams 256 rooms relentlessly pursuing you.
Available for **COMMODORE 64**. Available on **TAPE** or **DISK**. Played with **JOYSTICK**.

ILLUSTRATIONS: ELIZABETH HAUCK

*$3.00 additional for disk.

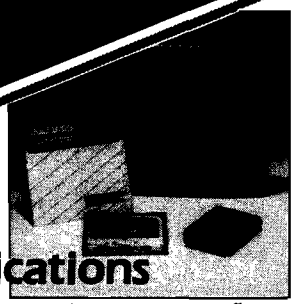
Check your LOCAL DEALER or order directly.
ORDERING: We accept personal checks, money orders, VISA, and MasterCard. Charge orders please include number and expiration date.
OVERSEAS ORDER: Please use charge, or have check payable through a U.S. bank.
CANADIAN CUSTOMERS: If you wish to write a check drawn through a Canadian bank, please multiply the total order by 1.25 for proper conversion.
Add $1.50 postage and handling per order. PA residents please add 6% sales tax.

VICTORY SOFTWARE INC.
7 Valley Brook Road
Paoli, Pennsylvania 19301
(215) 296-3787

VICTORY SOFTWARE

VISA

master charge



$149.95

Telecommunications with a difference!

Unexcelled communications power and compatibility, especially for professionals and serious computer users. Look us over; **SuperTerm** isn't just "another" terminal program. Like our famous Terminal-40, **it's the one others will be judged by.**

- **EMULATION**—Most popular terminal protocols: cursor addressing, clear, home, etc.
- **EDITING**—Full-screen editing of Receive Buffer
- **UP/DOWNLOAD FORMATS**—CBM, Xon-Xoff, ACK-NAK, CompuServe, etc.
- **FLEXIBILITY**—Select baud, duplex, parity, stopbits, etc. Even work off-line, then upload to system!
- **DISPLAY MODES**—40 column; 80/132 with side-scrolling
- **FUNCTION KEYS**—8 standard, 52 user-defined
- **BUFFERS**—Receive, Transmit, Program, and Screen
- **PRINTING**—Continuous printing with Smart ASCII interface and parallel printer; buffered printing otherwise
- **DISK SUPPORT**—Directory, Copy, Rename, Scratch

Options are selected by menus and EXEC file. Software on disk with special cartridge module. **Compatible with CBM and HES Automodems**; select ORIG/ANS mode, manual or autodial.

Write for the full story on SuperTerm; or, if you already want that difference, order today!

Requires: Commodore 64 or VIC-20, disk drive or Datasette, and compatible modem. VIC version requires 16K memory expansion. Please specify VIC or 64 when ordering.

Smart ASCII Plus . . . $59.95

The only interface which supports streaming —sending characters simultaneously to the screen and printer — with SuperTerm.

Also great for use with your own programs or most application programs, i.e., word processors. **Print modes:** CBM Graphics (w/many dot-addr printers), TRANSLATE, DaisyTRANSLATE, CBM/True ASCII, and PIPELINE.

Complete with printer cable and manual. On disk or cassette.

VIC 20 and Commodore 64 are trademarks of Commodore Electronics, Ltd.

MIDWEST MICRO inc.

DOSPLUS For C-64

By Michael Keryan

DOSPLUS — adds new utility functions to a Commodore 64 by use of the RESTORE key.

This article shows how to use the RESTORE key (which generates a Non-Maskable Interrupt) to access up to 32 new utility routines. Seven new functions are implemented, including versatile printer support, easy color changes, and a two-key append function that will allow you to combine two or more BASIC programs. The framework is provided to allow 25 more functions to be added.

The Commodore 64 is definitely one of the best values in home computer hardware. It is roughly comparable to IBM's PCjr at about one-fourth the cost. Why is it so inexpensive? For one thing, Commodore decided to use the same BASIC-in-ROM interpreter that was used in the VIC-20 and older PET machines. The KERNAL ROM was updated for the 64 but no

new BASIC commands were added to support the 64's outstanding sound, bit-mapped graphics, sprites, etc. To use these features you are left with a myriad of PEEKs and POKEs that in no way can be described as user-friendly. For example, to change the background screen color to black, you must type: POKE 53 281,0. To get a printed listing of a program, you type: OPEN 4,4: CMD 4: LIST, then afterwards: PRINT #4: CLOSE 4. An almost unbelievably complex series of POKES is necessary to program sprites and music.

A number of third-party software packages have attempted to solve some of these problems by adding new commands to BASIC, e.g. Ultra-BASIC 64. However, there is always some desirable function missing. For example, Ultra-BASIC 64 provides graphic screen dumps to the printer, but has no normal (alphanumeric) printer dump. Quite a few single-purpose machine language routines have appeared in magazines such as this recently which provide desirable add-on features. Invariably, these routines are placed either in the cassette buffer or at $C000, and require the SYS function to operate.

The most convenient way to access handy functions involves assigning certain keys or sequence of keys to the new functions. The DOS WEDGE (provided on the demo disk that comes with Commodore's 1541 disk drive) uses this technique. For example, instead of typing LOAD"$",8 then LIST, you type only $ to get a directory listing. In general, however, assignment of special keys makes your programs run slower since the absence or presence of these key sequences must continually be checked.

The Powerful RESTORE Key

This article provides a framework to add a series of extensions for the Commodore 64 using the RESTORE key, followed by another key to define the function. Normally when you hit the RESTORE key, you use it in conjunction with the RUN/STOP key, most likely to regain control of your computer when a program has gone into never-never land. However, hitting only RESTORE appears to have no effect — a virtually worthless key.

Actually, RESTORE is probably the most powerful key on the keyboard, and is ideally suited to controlling the computer. It provides a Non-Maskable

Listing 1

```
                        ; DOSPLUS   BY M. J. KERYAN
                        ;
                        ;     DUPLICATES NMI ROUTINE AT
                        ;     $FE43 IN ROM, CHECKS FOR
                        ;     KEY PRESS AFTER RESTORE KEY
                        ;
                        ;  ROM UTILITIES:
                        INIT1  = $E518
                        INIT2  = $FD15
                        INIT3  = $FDA3
                        STPFL  = $F6BC
                        CONTNM = $FE72
                        STOP   = $FFE1
                        GETIN  = $FFE4
                        ;
                        CIAIRQ = $DD0D; FOR RS/232 CHECK
                        TABL   = $CBC0; LOW BYTE POINTER
                        TABH   = $CBE0; HI  BYTE POINTER
                        ;
                        ORIGIN = $C900
                        * = ORIGIN
                        ;
C900  48        NEWNMI  PHA          ; SAVE REGISTERS
C901  8A                TXA
C902  48                PHA
C903  98                TYA
C904  48                PHA
C905  A9 7F             LDA #$7F
C907  8D 0D DD          STA CIAIRQ ; IRQ FLAG FOR
C90A  AC 0D DD          LDY CIAIRQ ;  RS/232
C90D  30 35             BMI RS232
C90F  20 BC F6          JSR STPFL  ; FLAG SET
C912  20 E1 FF          JSR STOP   ; STOP KEY?
C915  D0 0F             BNE NOTSTP ; NO, BRANCH
C917  20 15 FD  STOPKY  JSR INIT2  ; DUPLICATE NMI
C91A  20 A3 FD          JSR INIT3  ;  ROUTINES
C91D  20 18 E5          JSR INIT1
C920  20 47 C9          JSR INITNM ; NEW INITIAL
C923  6C 02 A0          JMP ($A002) ; WARM START
C926  58        NOTSTP  CLI          ; ENABLE KEYBD
C927  20 E4 FF  KEYLP   JSR GETIN  ; GET A KEY
C92A  C9 00             CMP #$00   ; WAIT/PRESSED
C92C  F0 F9             BEQ KEYLP
C92E  29 1F             AND #$1F   ; MASK FOR 32
C930  AA                TAX        ; UNIQUE KEYS
C931  BD C0 CB          LDA TABL,X ; GET LO BYTE
C934  8D 3E C9          STA *+10   ; SET UP JSR
C937  BD E0 CB          LDA TABH,X ;     HI BYTE
C93A  8D 3F C9          STA *+5
C93D  20 FF FF          JSR $FFFF  ; MODIFIED ADDR
C940  78                SEI        ; IRQ DISABLED
C941  68                PLA        ; GET BACK Y
C942  48                PHA
C943  A8                TAY
C944  4C 72 FE  RS232   JMP CONTNM
                        ;
                        ;INITIALIZATION ROUTINE FOR NMI
                        ;
                        NMINV  = $0318     ; KERNAL NMI VEC
                        IBSOUT = $0326     ; KERNAL CHROUT
                        BRDCLR = $D020
                        BCKCL1 = $D021
                        COLOR  = $0286     ; CURSOR COLOR
                        COLCH  = $00       ; CHARS = BLACK
                        COLBK  = $0C       ; BACKG = GREY
                        ;
C947  A9 00     INITNM  LDA #<NEWNMI ; RESTORE NMI
```

```
C949  8D 18 03          STA NMINV
C94C  A9 C9             LDA #>NEWNMI ; VECTOR TO
C94E  8D 19 03          STA NMINV+1  ; NEW ROUTINE
C951  A9 0C             LDA #COLBK
C953  8D 20 D0          STA BRDCLR   ; EDGE COLOR
C956  8D 21 D0          STA BCKCL1   ; BACKGROUND
C959  A9 00             LDA #COLCH
C95B  8D 86 02          STA COLOR    ; CHARACTERS
C95E  A9 71             LDA #<NEWOUT
C960  8D 26 03          STA IBSOUT
C963  A9 C9             LDA #>NEWOUT ; NEW OUTPUT
C965  8D 27 03          STA IBSOUT+1 ; VECTOR
C968  EA                NOP
C969  EA                NOP          ; FUTURE
C96A  EA                NOP          ; EXPANSION
C96B  A9 00             LDA #$00     ; PRINTER OFF
C96D  8D BB C9          STA PFLAG
C970  60                RTS
                ;
                ;NEW OUTPUT ROUTINE
                ;    VECTORS OUTPUT TO PRINTER IF
                ;    PFLAG IS SET IN ADDITION TO
                ;    CRT--DISK ACCESS WILL TURN
                ;    OFF PRINTER ACCESS
                ;
                CLRCHN = $FFCC ;KERNAL ROUTINES
                OLDOUT = $F1CA
                CLOSE  = $FFC3
                SETLFS = $FFBA
                SETNAM = $FFBD
                OPEN   = $FFC0
                CHKOUT = $FFC9
                ;
                VICMCR = $D018   ;VIC MEM CTRL REG
                FA     = $BA     ;CURRENT DEVICE #
                PFILE  = $7E     ;LOGICAL PTR FILE
                ;
C971  48        NEWOUT PHA
C972  AD BB C9          LDA PFLAG ;CHECK FLAG TO
C975  C9 04             CMP #$04  ;SEE IF PTR IS
C977  D0 3D             BNE RTNP  ;ACTIVE
C979  A5 BA             LDA FA    ;IF DEVICE IS
C97B  C9 08             CMP #$08  ;IS DISK THEN
C97D  F0 37             BEQ RTNP  ;DISABLE PRINTER
C97F  68                PLA
C980  8D BA C9          STA PTEMP
C983  48                PHA       ;SAVE REGISTERS
C984  8A                TXA
C985  48                PHA
C986  98                TYA
C987  48                PHA
C988  A9 7E             LDA #PFILE
C98A  20 C3 FF          JSR CLOSE ;FIRST CLOSE FILE
C98D  20 CA C9          JSR PCASE ;SET U/L CASE
C990  A9 7E             LDA #PFILE
C992  20 BA FF          JSR SETLFS
C995  A9 00             LDA #$00
C997  20 BD FF          JSR SETNAM
C99A  20 C0 FF          JSR OPEN
C99D  B0 0B             BCS PCLOSE ;CLOSE ON ERROR
C99F  A2 7E             LDX #PFILE
C9A1  20 C9 FF          JSR CHKOUT
C9A4  AD BA C9          LDA PTEMP
C9A7  20 CA F1          JSR OLDOUT ;CHAR TO PRINTER
C9AA  A9 7E     PCLOSE  LDA #PFILE
C9AC  20 C3 FF          JSR CLOSE
C9AF  20 CC FF          JSR CLRCHN ;RESET TO CRT
C9B2  68                PLA        ;RESTORE REGS
```

Interrupt (NMI) request to the computer's CPU. No matter what the computer is doing at the time, it will stop and jump to a special machine language program that determines why the interrupt was requested, and takes appropriate action.

The Commodore 64 NMI program basically does two things:

1. It determines if the NMI request came from the RS/232 port or from the RESTORE key. If the RS/232 port generated it, the data transfer is handled.
2. If the RESTORE key generated the interrupt, the NMI program checks to see if the STOP key was also pressed. If so, it jumps to initialization routines and then to a BASIC warmstart. If the STOP key wasn't pressed, the program has no noticeable effect.

So when you press the RESTORE key alone, nothing appears to happen, but actually the NMI program was executed. Commodore makes it very easy to replace their NMI program with one of your own, since the NMI routine vectors through RAM locations $0318-0319. Place the locations of your new NMI program in $0318-0319 and you've gained complete control of the computer through the RESTORE key.

Using the Program

An assembler program (Listing 1) illustrates how the RESTORE key can be used to add 32 new functions to the Commodore 64. To gain access to these new routines, press the RESTORE key followed by a second key that identifies the function. Seven new routines are provided in the Listing; twenty-five more are possible. The seven functions are as follows:

A to Append programs
B to change the Background color
C to change the Character color
E to change the Edge color
D to Dump the current screen to the printer
P to turn on the Printer
O to turn Off the printer.

Any other key pressed will return the computer to normal execution. Using the RUN/STOP and RESTORE keys together function as normally, except vectors are restored to the new routines.

Since this interrupt technique in-

volves no character checking during normal program execution, it won't slow down your programs. An unexpected benefit of using the RESTORE key is that it also acts as a PAUSE or NO-SCROLL key. When a program listing or data is scrolling by too fast to read, just hit RESTORE — this will freeze everything until you hit another key. To resume the scroll, press RETURN. Also, the RESTORE functions can be stacked. Suppose you just stopped a scroll by using RESTORE, and you would like a printout of the current screen. Just hit RESTORE a second time, followed by D to generate the screen copy. Then hit RETURN to continue scrolling. This is an obvious advantage over using STOP, etc.

The series of machine language routines reside in upper memory, protected from BASIC, Reset, etc., at locations $C800-CBFF. This is right next to the DOSWEDGE 5.1 utility program ($CC00-CFFF). Therefore the new routines (called DOSPLUS) can be loaded and initialized along with the WEDGE. Each routine is now described briefly.

Non-Maskable Interrupt Routine

The new NMI routine duplicates many of the functions of the original NMI routine located in ROM at $FE43. The CPU registers are saved and a check is made for an RS/232 interrupt. If the NMI request came from the RESTORE key, a check of the STOP key is made. If STOP was pressed at the same time, then three ROM initialization routines are run. A new initialization routine, INITNM, is then run (more on this later), followed by a warm start to BASIC.

Assuming you had only pressed the RESTORE key to access the new functions, the initializations and warm start are bypassed. The interrupt flag is cleared to allow keyboard entry. Then the program waits for a key to be pressed. The upper 3 bits of the ASCII value of the key are masked to zeroes, so only 32 unique keys are possible. Based on the value obtained above, pointers for the corresponding routines are set up into a JSR statement, then the JSR is taken, in which the new function is executed. After the subroutine is run, program flow returns to the old NMI routine in ROM which restores the registers and executes an RTI (return from interrupt) and normal program execution resumes.

```
C9B3  A8              TAY
C9B4  68              PLA
C9B5  AA              TAX
C9B6  68        RTNP  PLA
C9B7  4C CA F1        JMP OLDOUT ;RTS THRU OLDOUT
C9BA  0D        PTEMP .BYTE $0D
C9BB  00        PFLAG .BYTE $00
                      ;
                      TURN PRINTER ON/OFF
                      ;
C9BC  A9 04     PRNTON LDA #$04   ;ACTIVATE BY P
C9BE  85 BA           STA FA
C9C0  8D BB C9        STA PFLAG
C9C3  60              RTS
C9C4  A9 00     PRNTOF LDA #$00   ;ACTIVATE BY O
C9C6  8D BB C9        STA PFLAG
C9C9  60              RTS
                      ;
                      ;DETERMINE IF SCREEN IS UPPER OR
                      ;LOWER CASE--SET X AND Y
                      ;
C9CA  AD 18 D0  PCASE LDA VICMCR
C9CD  29 02           AND #$02
C9CF  F0 04           BEQ UPPER
C9D1  A0 07           LDY #$07
C9D3  D0 02           BNE PDEV
C9D5  A0 00     UPPER LDY #$00   ;Y IS SEC ADDR
C9D7  A2 04     PDEV  LDX #$04   ;X IS DEVICE #
C9D9  60               RTS
                      ;
                      ;DUMP SCREEN TO PRINTER
                      ;
                      DPNTR  = $FD      ;POINTER
                      ROWS   = $19      ;25 ROWS ON CRT
                      COLMNS = $28      ;40 COLUMNS
                      DPFILE = $7D      ;LOGICAL FILE #
                      ;
C9DA  20 CA C9  PDUMP JSR PCASE
C9DD  A9 7D           LDA #DPFILE
C9DF  20 BA FF        JSR SETLFS
C9E2  A9 00           LDA #$00
C9E4  20 BD FF        JSR SETNAM
C9E7  20 C0 FF        JSR OPEN
C9EA  B0 7A           BCS DPCLOS ;CLOSE ON ERROR
C9EC  A2 7D           LDX #DPFILE
C9EE  20 C9 FF        JSR CHKOUT
C9F1  A9 00           LDA #$00
C9F3  85 FD           STA DPNTR
C9F5  A9 04           LDA #$04
C9F7  85 FE           STA DPNTR+1
C9F9  20 89 CA        JSR PCR     ;CAR RET
C9FC  20 7E CA        JSR LINE    ;SYMBOL LINE
C9FF  20 97 CA        JSR LINBLK  ;BLANK LINE
CA02  A2 19           LDX #ROWS
CA04  20 E1 FF  ROLOOP JSR STOP   ;IF STOP KEY,
CA07  F0 5D           BEQ DPCLOS ;THEN STOP PRINT
CA09  20 75 CA        JSR SPC16  ;MARGIN
CA0C  20 91 CA        JSR BOXSYM ;SYMBOL
CA0F  20 8D CA        JSR PSPACE ;SPACE
CA12  A0 00           LDY #$00
CA14  B1 FD     COLOOP LDA (DPNTR),Y ;GET CRT CHAR
CA16  C9 22           CMP #$22    ;IF QUOTE,
CA18  D0 02           BNE SHFTQT ;CHANGE TO
CA1A  A9 27           LDA #$27    ;APOSTROPHE
CA1C  C9 A2     SHFTQT CMP #$A2   ;IF SHIFT QUOTE,
CA1E  D0 02           BNE NOTQT   ;CHANGE TO
CA20  A9 A7           LDA #$A7    ;SHIFT APOSTROPHE
CA22  8D BA C9  NOTQT STA PTEMP
```

NMI Initialization Routine

As mentioned above, a combination RUN/STOP and RESTORE restores the peripheral chips, colors, etc. back to their default values. This means we would lose control of the NMI routine because the KERNAL ROM addresses are reinstated in the indirect vectors in RAM. The new initialization routine therefore places our addresses back into the vectors for the NMI routine and the OUTPUT routine (used in the printer utilities).

The initialization routine also allows us to change the colors to new default values, in this case to black characters on a grey background. If you desire different default colors, change them here.

Lastly, the printer flag is reset to disable the printer, if it was on before the intialization was run.

New OUTPUT Routine

NEWOUT allows you to turn on the printer (device #4) so that all output to the screen can go to both the screen and the printer. When this routine is activated, to get a program listing, just type LIST.

The NEWOUT routine first checks to see if the printer flag is off. Then it checks to see if the active device is #8. Disk access is set up to kill the printer output to avoid possible contention problems on the serial bus.

The program then checks to see if the screen is in upper/graphics or lower/upper case. A file is then opened to device #4 with a secondary address of 0 or 7, depending on the case.

The character is sent to the printer, the file is closed, then the character is sent to the screen. Note that the printer file is opened and closed for each character. The printer output mode is turned on by RESTORE, P, and turned off by RESTORE, O.

Dump Screen to Printer

In addition to the printer on/off functions, a screen dump routine is provided. This is accessed by the RESTORE key followed by a D. The entire 40 × 25 screen is copied character by character to the printer. The printed output is indented to center the text on the paper and a dotted outline is drawn around the 40 × 25 matrix. As before, the printed output is made the same case as the screen.

The PDUMP routine opens and

```
CA25  29 80        AND #$80    ;IF REVERSE CHAR
CA27  8D AF CA     STA REVFLG  ;THEN SEND REV
CA2A  F0 05        BEQ DECODE  ;CODE TO PRINTER
CA2C  A9 12        LDA #$12    ;
CA2E  20 CA F1     JSR OLDOUT
CA31  AD BA C9  DECODE LDA PTEMP   ;CHANGE SCREEN
CA34  29 3F        AND #$3F    ;CODE TO ASCII
CA36  0E BA C9     ASL PTEMP   ;CODE FOR
CA39  2C BA C9     BIT PTEMP   ;PRINTER
CA3C  10 02        BPL *+4
CA3E  09 80        ORA #$80
CA40  70 02        BVS *+4
CA42  09 40        ORA #$40
CA44  20 CA F1     JSR OLDOUT  ;CHAR TO PRINTER
CA47  AD AF CA     LDA REVFLG  ;IF REVERSE TURN
CA4A  F0 05        BEQ CHRSNT  ;OFF REV MODE
CA4C  A0 92        LDA #$92
CA4E  20 CA F1     JSR OLDOUT
CA51  C8        CHRSNT INY          ;NEXT COLUMN
CA52  C0 28        CPY #COLMNS
CA54  D0 BE        BNE COLOOP
CA56  20 A5 CA     JSR LNEND   ;END OF LINE
CA59  98           TYA
CA5A  18           CLC
CA5B  65 FD        ADC DPNTR
CA5D  85 FD        STA DPNTR
CA5F  90 02        BCC *+4
CA61  E6 FE        INC DPNTR+1 ;REVISE POINTER
CA63  CA           DEX
CA64  C0 9E        BNE ROLOOP
CA66  20 97 CA  DPCLOS JSR LINBLK ;DONE WITH CRT
CA69  20 7E CA     JSR LINE
CA6C  20 CC FF     JSR CLRCHN  ;RESET TO CRT
CA6F  A9 7D        LDA #DPFILE
CA71  20 C3 FF     JSR CLOSE
CA74  60           RTS         ;DONE WITH DUMP
                   ;
                   ;SUBROUTINES TO SUPPORT DUMP
                   ;
CA75  A0 10     SPC16  LDY #$10    ;16 BLANKS
CA77  20 8D CA  SP16LP JSR PSPACE
CA7A  88           DEY
CA7B  D0 FA        BNE SP16LP
CA7D  60           RTS
                   ;
CA7E  20 75 CA  LINE   JSR SPC16   ;LINE OF SYMBOLS
CA81  A0 2C        LDY #$2C
CA83  20 91 CA  LINELP JSR BOXSYM
CA86  88           DEY
CA87  D0 FA        BNE LINELP
CA89  A9 0D     PCR    LDA #$0D    ;CARR RETURN
CA8B  D0 06        BNE OUTCHR
CA8D  A9 20     PSPACE LDA #$20    ;SPACE
CA8F  D0 02        BNE OUTCHR
CA91  A9 2E     BOXSYM LDA #$2E    ;PERIOD
CA93  20 CA F1  OUTCHR JSR OLDOUT
CA96  60           RTS
                   ;
CA97  20 75 CA  LINBLK JSR SPC16   ;BLANK LINE
CA9A  20 91 CA     JSR BOXSYM
CA9D  A0 29        LDY #$29
CA9F  20 8D CA  LBLKLP JSR PSPACE
CAA2  88           DEY
CAA3  D0 FA        BNE LBLKLP
CAA5  20 8D CA  LNEND  JSR PSPACE
CAA8  20 91 CA     JSR BOXSYM
CAAB  20 89 CA     JSR PCR
CAAE  60           RTS
```

```
                        ;
CAAF  00                REVFLG .BYTE $00 ;ZERO IF NOT REV
                        ;
                        ;CHANGE SCREEN COLORS
                        ;3 ENTRY POINTS FOR EDGE, BACKGROUND,
                        ;AND CHARACTERS--ALL 3 CAN BE
                        ;CHANGED WITH ONE ENTRY
                        ;
CAB0  A2 00             EDGE    LDX #$00      ;EDGE ENTRY
CAB2  F0 02                     BEQ CHGCOL
CAB4  A2 01             BACKGD  LDX #$01      ;BACKG ENTRY
CAB6  FE 20 D0          CHGCOL  INC BRDCLR,X  ;CHANGE COLOR
CAB9  20 E4 FF          WAITCL  JSR GETIN     ;GET KEY
CABC  C9 00                     CMP #$00
CABE  F0 F9                     BEQ WAITCL
CAC0  C9 42                     CMP #$42      ;B KEY
CAC2  F0 F0                     BEQ BACKGD
CAC4  C9 45                     CMP #$45      ;E KEY
CAC6  F0 E8                     BEQ EDGE
CAC8  C9 43                     CMP #$43      ;C KEY
CACA  F0 01                     BEQ CHARAC
CACC  60                        RTS           ;ANOTHER KEY
CACD  A2 FA             CHARAC  LDX #$FA
CACF  FE FF D7          FILLCH  INC $D7FF,X   ;CHANGE ALL
CAD2  FE F9 D8                  INC $D8F9,X   ;OF COLOR
CAD5  FE F3 D9                  INC $D9F3,X   ;MEMORY
CAD8  FE ED DA                  INC $DAED,X
CADB  CA                        DEX
CADC  D0 F1                     BNE FILLCH
CADE  EE 86 02                  INC COLOR     ;CURSOR COLOR
CAE1  18                        CLC
CAE2  90 D5                     BCC WAITCL    ;CONTINUE
                        ;
                        ;APPEND PROGRAM
                        ;TO TACK ON SUBROUTINE PROGRAMS
                        ;TO OTHER PROGRAMS
                        ;
                        TXTTAB = $2B
                        VARTAB = $2D
                        ;
CAE4  4C E7 CA          APPEND  JMP APPDON  ;MODIFIABLE JMP
CAE7  A5 2C             APPDON  LDA TXTTAB+1
CAE9  8D 3F CB                  STA APHI
CAEC  A5 2B                     LDA TXTTAB
CAEE  8D 40 CB                  STA APLO
CAF1  A5 2E                     LDA VARTAB+1
CAF3  85 2C                     STA TXTTAB+1
CAF5  38                        SEC
CAF6  A5 2D                     LDA VARTAB
CAF8  E9 02                     SBC #$02
CAFA  85 2B                     STA TXTTAB
CAFC  B0 02                     BCS APON
CAFE  C6 2C                     DEC TXTTAB+1
CB00  A9 2A             APON    LDA #<APPDOF
CB02  8D E5 CA                  STA APPEND+1
CB05  A9 CB                     LDA #>APPDOF
CB07  8D E6 CA                  STA APPEND+2
CB0A  20 41 CB                  JSR MESSAG
CB0D  0D 4C 4F 41               .BYTE $0D,$4C,$4F,$41
CB11  44 20 50 47               .BYTE $44,$20,$50,$47
CB15  4D 32 2C 20               .BYTE $4D,$32,$2C,$20
CB19  48 49 54 20               .BYTE $48,$49,$54,$20
CB1D  52 45 53 54               .BYTE $52,$45,$53,$54
CB21  4F 52 45 2C               .BYTE $4F,$52,$45,$2C
CB25  20 41 0D 00               .BYTE $20,$41,$0D,$00
CB29  60                        RTS
CB2A  AD 3F CB          APPDOF  LDA APHI
CB2D  85 2C                     STA TXTTAB+1
```

closes a file similarly to the OUTPUT routine. A big difference is the transformation from screen code to ASCII code. Commodore does some very strange things to the printer and screen when a reverse character follows a quotation mark. It is very difficult to get an exact screen dump when this occurs. A quotation mark is replaced by an apostrophe on printed output, to allow reverse characters to be printed (if your printer or printer/interface will handle them). Examples of screen dumps are shown in Figure 1. In this case the printer used (NEC 8023/Tymac Connection Interface) does not provide reverse alphanumerics; a Commodore 1525 printer would have printed the directory header with reverse characters.

Change Screen Colors

A routine is provided to change the screen colors. This routine has three entry points. RESTORE, B changes the background color, RESTORE, E changes the edge color, and RESTORE, C changes the character color. Note that all characters on the screen will be changed, not just the characters following the cursor. When the colors are changed, their code number is incremented by one, from 0 to 15, back to 0, etc.

Once any one color has been changed, you can continue to step through the colors by repetitively pressing the same key (no need to hit RESTORE again). In addition, no matter which one you started with (B, E, or C), you can change the others by pressing the corresponding key. Any other key will get you out of the color change mode.

Append

To append a program (normally consisting of subroutines or DATA statements) to another, first load the main program in the usual manner: LOAD"PRG1",8. Then press RESTORE, A This raises the bottom of the BASIC program workspace to the end of the first program. The computer than prompts you to load the second program, e.g. LOAD"PRG2",8 and press RESTORE, A again. This time the bottom of the BASIC workspace is moved back down to the previous location. Your two programs are now one, which you should SAVE with a new name. You can continue to append more programs using the same technique. One word of caution — make sure that the ap-

pended programs have higher line numbers than the main programs since they are not merged — only appended.

A general purpose message printer routine is used above. To use this subroutine in your machine language routines, do a JSR to MESSAG followed by your message in ASCII characters. End the message table with a zero. After the message is printed, control will return to the next byte following the zero. In addition to normal text, cursor controls, color changes, etc. can be intermixed with text for special effects.

Entering the Program

The DOSPLUS routines currently start at $C800 and extend through $CBFF. Space for more routines is available from $C807-C8FF and from $CB60-CBBE. Additional extensions can be assembled below $C800. The DOS WEDGE occupies $CC00-CFFF, so it is advantageous to load and initialize the DOS WEDGE and DOSPLUS at the same time.

There are several ways to get the program into your Commodore 64. If you have an Assembler, type in the Assembly code as shown in Listing 1. A compact BASIC loader is also provided; see Listing 2. Checksums are used to catch your typing errors, which will terminate the program. Note the change to line #1260 required if you do not intend to run the BASIC loader with DOS 5.1 in memory. A third method is to send the author $10 (US) for a disk containing the programs shown in Figure 1. (For foreign requests, please provide sufficient return postage). If you load and run the first program DOSPLUS, both the DOS-WEDGE and the RESTORE routines will be loaded and initiated. Then type NEW and you're ready to go.

Seven new functions were added, but the tables will allow a total of 32. Some possible additional functions include variable listing, single step, trace, autoline, hex-decimal converter, save & swap screens (for animation), etc. Don't hesitate to write if you have a useful addition. Or better yet, write it up for all MICRO readers.

MICRO

You may contact the author:

Michael J. Keryan
713 Locust Drive
Tallmadge, Ohio 44278

```
CB2F  AD 40 CB        LDA APLO
CB32  85 2B           STA TXTTAB
CB34  A9 E7     APOFF LDA #<APPDON
CB36  8D E5 CA        STA APPEND+1
CB39  A9 CA           LDA #>APPDON
CB3B  8D E6 CA        STA APPEND+2
CB3E  60              RTS
CB3F  FF        APHI  .BYTE $FF
CB40  FF        APLO  .BYTE $FF
                      ;
                      ;MESSAGE PRINTER
                      ;JSR MESSAG--FOLLOW WITH ASCII
                      ;MESSAGE--END WITH ZERO
                      ;
                      CHROUT = $FFD2
                      POINMS = $FB
                      ;
CB41  68        MESSAG PLA           ;SAVE POINTERS
CB42  85 FB           STA POINMS     ;FOR RETURN
CB44  68              PLA
CB45  85 FC           STA POINMS+1
CB47  A2 00           LDX #$00
CB49  E6 FB     MSSGLP INC POINMS    ;PARSE MESSAGE
CB4B  D0 02           BNE *+4
CB4D  E6 FC           INC POINMS+1 ;BUMP POINTER
CB4F  A1 FB           LDA (POINMS,X) ;GET CHAR
CB51  F0 06           BEQ MSSGBK     ;END ON ZERO
CB53  20 D2 FF        JSR CHROUT     ;PRINT IT
CB56  18              CLC
CB57  90 F0           BCC MSSGLP     ;CONTINUE
CB59  A5 FC     MSSGBK LDA POINMS+1
CB5B  48              PHA
CB5C  A5 FB           LDA POINMS
CB5E  48              PHA
CB5F  60              RTS
                      ;
                      * = $C800
                      INIDOS = $CC00
                      ;
C800  20 47 C9  BASINT JSR INITNM ;INIT NMI UTILS
C803  20 00 CC        JSR INIDOS ;INIT DOS WEDGE
C806  60              RTS
                      ;
                      * = $CBBF
CBBF  60        RETRN .BYTE $60
                      ;
                      * = $CBC0
                      TABL = *
CBC0  BF E4 B4 CD     .BYTE $BF,$E4,$B4,$CD
CBC4  DA B0 BF BF     .BYTE $DA,$B0,$BF,$BF
CBC8  BF BF BF BF     .BYTE $BF,$BF,$BF,$BF
CBCC  BF BF BF C4     .BYTE $BF,$BF,$BF,$C4
CBD0  BC BF BF BF     .BYTE $BC,$BF,$BF,$BF
CBD4  BF BF BF BF     .BYTE $BF,$BF,$BF,$BF
CBD8  BF BF BF BF     .BYTE $BF,$BF,$BF,$BF
CBDC  BF BF BF BF     .BYTE $BF,$BF,$BF,$BF
                      ;
                      * = $CBE0
                      TABH = *
CBE0  CB CA CA CA     .BYTE $CB,$CA,$CA,$CA
CBE4  C9 CA CB CB     .BYTE $C9,$CA,$CB,$CB
CBE8  CB CB CB CB     .BYTE $CB,$CB,$CB,$CB
CBEC  CB CB CB C9     .BYTE $CB,$CB,$CB,$C9
CBF0  C9 CB CB CB     .BYTE $C9,$CB,$CB,$CB
CBF4  CB CB CB CB     .BYTE $CB,$CB,$CB,$CB
CBF8  CB CB CB CB     .BYTE $CB,$CB,$CB,$CB
CBFC  CB CB CB CB     .BYTE $CB,$CB,$CB,$CB
                      .END
```

Listing 2

```
  1 PRINT"[CLR]            DOSPLUS LOADER":PRINT"            WAIT--LOADING"
 10 DIM N(1 0): I=0
 20 READ A$: I=I+1: CS=0
 30 X$=LEFT$(A$,2): GOSUB 100: IF Z<1 THEN SYS 5120 0: END
 40 BY=Z: KS=Z: X$=MID$(A$,3,4): GOSUB 100: LC=Z
 50 X$=MID$(A$,3,2): GOSUB 1 00: KS=KS+Z: X$=MID$(A$,5,2): GOSUB 100: KS=KS+Z
 60 FOR J=1TOBY: X$=MID$(A$,5+J*2 ,2): PRINT".";
 70 GOSUB 100: N(J)=Z: CS=CS+Z: NEXTJ
 80 X$=MID$(A$,7+BY*2,4): GOSUB 100: IF Z<>CS+KS THEN PRINT"ERROR LINE #";I:
 STOP
 90 FOR J=1TOBY: POKE LC+J-1,N(J ): NEXTJ:PRINT"*": GOTO 20
100 :
110 REM HEX TO DECIMAL
120 REM X$ IS HEX NUMBER TO  BE CONVERTED
130 REM Z IS DECIMAL NUMBER CONVERTED
140 Z=0
150 L=LEN(X$): FORK=1TO L
160 Y=ASC(MID$(X$,K,1))
190 Z=Z*16+Y-48+7*(Y>57)
200 NEXTK: RETURN
1000 DATA"18C900488A489848A97F8D0DDDAC0DDD303520BCF620E1FFD00F200C46"
1010 DATA"18C91815FD20A3FD2018E52047C96C02A05820E4FFC900F0F9291F0C7B"
1020 DATA"18C930AABDC0CB8D3EC9BDE0CB8D3FC920FFFF786848A84C72FEA90FE7"
1030 DATA"18C948008D1803A9C98D1903A90C8D20D08D21D0A9008D8602A9710A74"
1040 DATA"18C9608D2603A9C98D2703EAEAEAA9008DBBC96048ADBBC9C904D00E0E"
1050 DATA"18C9783DA58AC908F037688DBAC9488A489848A97E20C3FF20CAC90E1B"
1060 DATA"18C990A97E20BAFFA90020BDFF20C0FFB00BA27E20C9FFADBAC9200EE8"
1070 DATA"18C9A8CAF1A97E20C3FF20CCFF68A868AA684CCAF10D00A90485BA0EC2"
1080 DATA"18C9C08DBBC960A9008DBBC960AD18D02902F004A007D002A000A20C9B"
1090 DATA"18C9D8046020CAC9A97D20BAFFA90020BDFF20C0FFB07AA27D20C90E65"
1100 DATA"18C9F0FFA90085FDA90485FE2089CA207ECA2097CAA21920E1FFF00F32"
1110 DATA"18CA085D2075CA2091CA208DCAA000B1FDC922D002A927C9A2D0020CB0"
1120 DATA"18CA20A9A78DBAC929808DAFCAF005A91220CAF1ADBAC9293F0EBA0DFC"
1130 DATA"18CA38C92CBAC9100209807002094020CAF1ADAFCAF005A99220CA0C03"
1140 DATA"18CA50F1C8C028D0BE20A5CA981865FD85FD9002E6FECAD09E20970FE9"
1150 DATA"18CA68CA207ECA20CCFFA97D20C3FF60A010208DCA88D0FA6020750E3D"
1160 DATA"18CA80CAA02C2091CA88D0FAA90DD006A920D002A92E20CAF160200D1E"
1170 DATA"18CA9875CA2091CAA029208DCA88D0FA208DCA2091CA2089CA60000D8B"
1180 DATA"18CAB0A200F002A201FE20D020E4FFC900F0F9C942F0F0C945F0E8103D"
1190 DATA"18CAC8C943F00160A2FAFEFFD7FEF9D8FEF3D9FEEDDACAD0F1EE8614D4"
1200 DATA"18CAE0021890D54CE7CAA52C8D3FCBA52B8D40CBA52E852C38A52D0C9C"
1210 DATA"18CAF8E902852BB002C62CA92A8DE5CAA9CB8DE6CA2041CB0D4C4F0DAD"
1220 DATA"18CB1041442050474D322C2048495420524553544F52452C20410D06BD"
1230 DATA"18CB280060AD3FCB852CAD40CB852BA9E78DE5CAA9CA8DE6CA60FF0F16"
1240 DATA"18CB40FF6885FB6885FCA200E6FBD002E6FCA1FBF00620D2FF18901055"
1250 DATA"08CB58F0A5FC48A5FB4860064C"
1260 DATA"07C8002047C9602000CC034B"
1270 DATA"18CB8F60BFE4B4CDDAB0BFBFBFBFBFBFBFBFBFC4BCBFBFBFBFBFBFBF1361"
1280 DATA"18CBD7BFBFBFBFBFBFBFBFBFBFCBCACACAC9CACBCBCBCBCBCBCBCBCB1450"
1290 DATA"11CBEFC9C9CBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCBCB0F42"
1300 DATA"00"
1500 :
1510 REM IF DOS WEDGE WILL NOT BE RUN WITH THIS PROGRAM, REPLACE LINE 1260
     WITH:
1520 REM 1260 DATA"04C8002047C960025C"
```

MICRO™

SANYO MONITOR SALE!!

9" Data Monitor

- 80 Columns x 24 lines
- Green text display
- East to read - no eye strain
- Up front brightness control
- High resolution graphics
- Quick start - no preheating
- Regulated power supply
- Attractive metal cabinet
- UL and FCC approved

- *15 Day Free Trial - 90 Day Immediate Replacement Warranty*

9" Screen - Green Text Display	**$ 79.00**
12" Screen - Green Text Display (anti-reflective screen)	**$ 99.00**
12" Screen - Amber Text Display (anti-reflective screen)	**$119.00**
14" Screen - Color Monitor (national brand)	**$249.00**

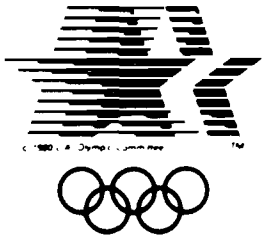
Display Monitors From Sanyo

With the need for computing power growing every day, Sanyo has stepped in to meet the demand with a whole new line of low cost, high quality data monitors. Designed for commercial and personal computer use. All models come with an array of features, including up-front brightness and contrast controls. The capacity 5 x 7 dot characters as the input is 24 lines of characters with up to 80 characters per line.

Equally important, all are built with Sanyo's commitment to technological excellence. In the world of Audio/Video, Sanyo is synonymous with reliability and performance. And Sanyo quality is reflected in our reputation. Unlike some suppliers, Sanyo designs, manufactures and tests virtually all the parts that go into our products, from cameras to stereos. That's an assurance not everybody can give you!

SANYO

Official Video Products of the Los Angeles 1984 Olympics

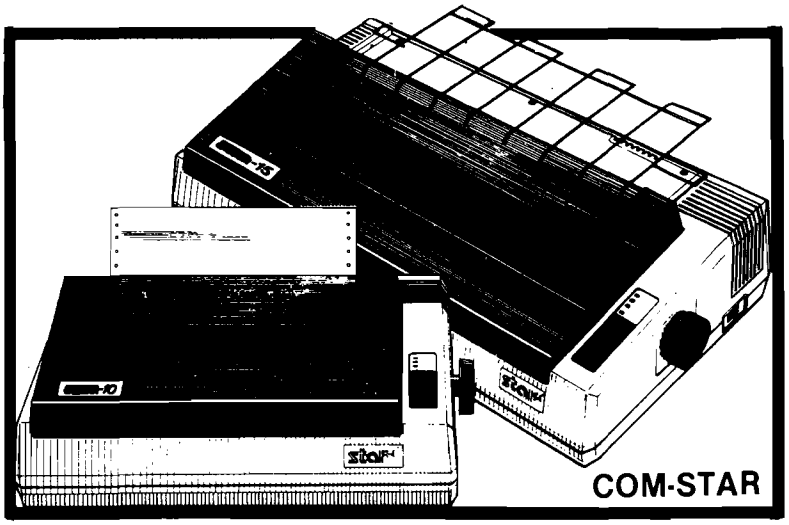


• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order



FANTASTIC PRINTER SALE

as low as **$149.00**

- *15 Day Free Trial - 180 Day Immediate Replacement Warranty*

	LIST	SALE
80 COLUMN THERMAL PRINTER — 60 CPS Bi-directional, dot matrix, prints 8½" letter size paper, full 80 columns, high resolution graphics, dot bit addressable, special symbols and true decenders! (Centronics parallel interface)	$199	**$149**
80 COLUMN TRACTOR-FRICTION PRINTER — 80 CPS Bi-directional, dot matrix, impact, prints single sheets, continuous feed paper, adjustable columns, 40 to 132 characters! Roll paper adapter $32.95. (Serial or Centronics parallel interface)	$399	**$209**
PREMIUM QUALITY 10" CARRIAGE T/F PRINTER — 120 CPS Bi-directional, impact, 9 x 9 dot matrix with double strike for 18 x 18 dot matrix. High resolution bit image (120 x 144 dot matrix) underlining back spacing, left and right margin settings, true lower decenders, with super and sub scripts. Prints standard, italic, block graphics, special characters, plus 24 of user definable characters and much more!! Prints single sheets, continuous feed and roll paper! (Centronics parallel interface)	$499	**$289**
PREMIUM QUALITY 15½" CARRIAGE PRINTER — 120 CPS Has all the features of the Premium Quality 10" Carriage T/F Printer above plus a 15½" carriage and more powerful electronic components to handle large business forms! (Centronics parallel interface)	$599	**$379**
HIGH SPEED PREMIUM QUALITY T/F 10" PRINTER — 160 CPS Save printing time with these plus features: 160 CPS speed, 100% duty cycle, 8K buffer diverse character fonts special symbols and true decenders, vertical and horizontal tabs. This is Red Hot Efficiency!!! (Serial or Centronics parallel interface)	$699	**$499**
HIGH SPEED PREMIUM QUALITY T/F 15½" PRINTER — 160 CPS Has all the features of the 10" Carriage high speed printer plus a 15½" carriage and more powerful electronics to handle larger business forms! (Serial or Centronics parallel interface)	$799	**$599**

PARALLEL PRINTER INTERFACES: (IN STOCK)

- For VIC-20 and COMMODORE 64 **$49.00**
- For all APPLE COMPUTERS **$69.00**
- For ATARI 400 and 800 COMPUTERS **$79.00**

NOTE: *Other printer interfaces are available at computer stores!*

SALE SALE

WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

FLOPPY DISKS SALE *$1.19 ea.

Economy Model or Cadillac Quality

LORAN CERTIFIED PERSONAL COMPUTER DISK ***We have the lowest prices!*** LORAN CERTIFIED PERSONAL COMPUTER DISK

***ECONOMY DISKS**

Good quality 5¼" single sided single density with hub rings.

Bulk Pac	100 Qty.	$1.19 ea.	Total Price	$119.00
	10 Qty.	1.39 ea.	Total Price	13.90

CADILLAC QUALITY

• Each disk certified • Free replacement lifetime warranty • Automatic dust remover

For those who want cadillac quality we have the Loran Floppy Disk. Used by professionals because they can rely on Loran Disks to store important data and programs without fear of loss! Each Loran disk is 100% certified (an exclusive process) plus each disk carries an exclusive FREE REPLACEMENT LIFETIME WARRANTY. With Loran disks you can have the peace of mind without the frustration of program loss after hours spent in program development.

100% CERTIFICATION TEST

Some floppy disk manufacturers only sample test on a batch basis the disks they sell, and then claim they are certified. Each Loran disk is individually checked so you will never experience data or program loss during your lifetime!

FREE REPLACEMENT LIFETIME WARRANTY

We are so sure of Loran Disks that we give you a free replacement warranty against failure to perform due to faulty materials or workmanship for as long as you own your Loran disk.

AUTOMATIC DUST REMOVER

Just like a record needle, disk drive heads must travel hundreds of miles over disk surfaces. Unlike other floppy disks the Loran smooth surface finish saves disk drive head wear during the life of the disk. (A rough surface will grind your disk drive head like sandpaper). The lint free automatic CLEANING LINER makes sure the disk-killers (dust & dirt) are being constantly cleaned while the disk is being operated. PLUS the Loran Disk has the highest probability rate of any other disk in the industry for storing and retaining data without loss for the life of the disk.

Loran is definitely the Cadillac disk in the world

Just to prove it even further, we are offering these super LOW INTRODUCTORY PRICES

List $4.99 ea. INTRODUCTORY SALE PRICE $2.99 ea. (Box of 10 only) Total price $29.90

$3.33 ea. (3 quantity) Total price $9.99

All disks come with hub rings and sleeves in an attractive package.

DISK DRIVE CLEANER $19.95

Everyone needs a disk drive doctor

FACTS

- 60% of all drive downtime is directly related to poorly maintained drives.
- Drives should be cleaned each week regardless of use.
- Drives are sensitive to smoke, dust and all micro particles.
- Systematic operator performed maintenance is the best way of ensuring error free use of your computer system.

The Cheetah disk drive cleaner can be used with single or double sided 5¼" disk drives. The Cheetah is an easy to use fast method of maintaining efficient floppy diskette drive operation.

The Cheetah cleaner comes with 2 disks and is packed in a protective plastic folder to prevent contamination.

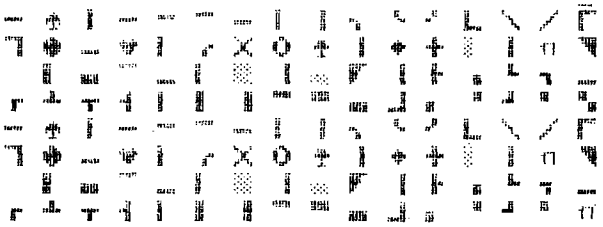
List $29.95 / Sale $19.95

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

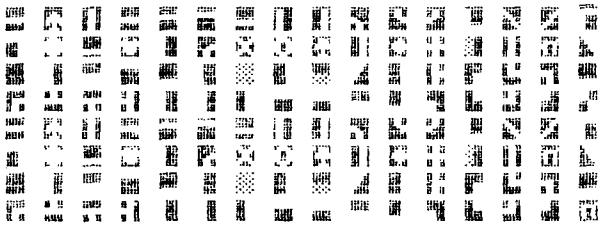
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa · MasterCard · C.O.D.

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS!)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Sample Commodore Graphic Character Set for the FX80.



Normal Characters



Reversed Characters

Generating Characters for the EPSON FX80 on a Commodore 64

by **Robert M. Tripp**

Basic Printing Concepts

The evolution of plain paper printers has been toward less expensive units that do more. Those who have entered the microcomputer revolution lately may not realize that it was only five or six years ago that a used Teletype (tm) at over $500 was a hot item. And all it could do was produce numbers, punctuation and upper case alphabetics - at 10 character per second with a lot of noise and it was not very transportable! Printers have come a long way since then and are getting better all the time. One of the most useful features to come along recently is the ability for the user to define his own character set. Until this capability was added, the user had three choices if he needed characters that the printer did not support:

1. Find some way to live within the constraints of the supported character set, such as using [CLR] to stand for the inverse heart shape that Commodore computers use as the displayable representation of the clear screen function.
2. Print the character information in a graphics mode.
3. Purchase a custom printer with the appropriate character set - normally limiting the user to the printer manufactured by the microcomputer company.

Each alternatives has a major drawback. The first requires that special programs be written to do the conversion from the microcomputer display character code to the alphanumeric string that will represent it, and results in a printout that does not represent the microcomputer display very well. The second requires a program to convert the character oriented information into a graphic form, and due to the fact that information has to be transmitted for every bit on the line, is quite slow. The third is an added expense and may result in a printer with limited capabilities and few, if any, of the more sophisticated printer features.

A very neat solution has been provided recently by the general purpose printer manufacturers. Many now include the ability for the user to define some or all of the character set in the printer's RAM. While the number of characters that may be so defined, the matrix for generating the characters, and the exact protocol for defining the characters varies from printer to printer, the basic concepts are the same. MICRO, as a microcomputer user, probably has more need for the user defineable characters than the average user since we deal so often with printed material and we deal with such a variety of microcomputers. We investigated the available printers and selected the Epson FX80 since it has allows all 256 characters to be user defined. Many of the other printers with UDCs currently available only permit a portion of the character set to be redefined and some do not support copying the existing ROM set into RAM to form the basis for the UDC. These two factors may not be significant to you, and other printers may be better for your particular requirements. Most of the material in this article will be applicable to printers in general.

Custom Character Sets

The **GRAPHICS** issue of MICRO (#66, November 1983) discussed character oriented graphic characters and provided methods for generating character sets on the Commodore 64, VIC 20 and Atari microcomputers. The examples used in the articles were, for the most part, generated on our FX80. A few of the possible uses for the UDC include:

1. Graphic characters that replicate those of the micro's display.
2. Custom graphic characters such as those used for line drawings, macro characters, special symbols, and so forth.
3. Alphabets for foreign languages, computer readable codes such as bar codes, optical scanner characters, special fonts such as the magnetic ink characters found on checks, etc.
4. Character segments that may be used to form 'super sized' printouts for displays such as calendars, printed material for the visually handicaped, an so on.

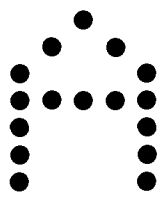
Generation of User Defineable Characters

The FX80 manual has a very good discussion of the details of generating user defineable characters. This presentation

```
  1 REM  "FXPCG GENERATOR"
 10 PRINT"□GENERATING CHARACTERS"
 15 CLR:GOSUB 5000:
    REM GENERATE CHARACTERS
 20 GOSUB 5100: REM DEFINE EQUATES
 30 GOSUB 4000: REM SETUP SCREEN
 40 GOSUB 1000:GOTO 2000: REM EDIT
1000 REM  FX CHARACTER DISPLAY
1010 BY=FX+13*CH:PRINT"S";
1020 FOR J=7 TO 0 STEP -1:K=2^J
1030 FOR L=1 TO 11 STEP 2:I=BY+L
1040 IF PEEK(I) AND K THEN 1100
1050 IF PEEK(I-1) AND K THEN 1080
1060 IF PEEK(I+1) AND K THEN 1110
1070 PRINT ".";:GOTO 1200
1080 IF PEEK(I+1) AND K THEN 1120
1090 PRINT CHR$(LH);:GOTO 1200
1100 PRINT CHR$(CL);:GOTO 1200
1110 PRINT CHR$(RH);:GOTO 1200
1120 PRINT CHR$(LR);
1200 NEXT
1210 PRINT:NEXT
1220 RETURN
1400 REM  OUTPUT FX CHARACTERS TO MEMORY
1410 BY=FX+13*CH
1420 FOR L=0 TO 12:POKE(BY+L),0:NEXT
1430 FOR L=1 TO 11 STEP 2:I=BY+L:A=0:B=0:
     XX=INT(L/2)
1440 FOR J=7 TO 0 STEP -1:K=2^J
1450 X=PEEK(OG+XX)
1460 IF X=CX THEN A=A OR K
1470 IF X=RX OR X=DX THEN B=B OR K
1480 XX=XX+RW:NEXT
1490 POKE(I),A:POKE(I+1),B:NEXT
1500 RETURN
2000 REM EDIT
2010 H=0:V=0:CP=87
2015 PRINT"S]]]]]]]]CHAR #";CH;"    S";
2020 PO=OG+V*RW+H:X=PEEK(PO)
2022 Y=PEEK(PO+1):BY=FX+CH*11+V
2025 POKE PO,CP
2030 GET T$:IF T$="" THEN 2030
2040 IF T$="£" THEN CP=225: GOTO 2025
2050 IF T$="^" THEN CP=87: GOTO 2025
2060 IF T$=CL$ THEN H=H+1:GOTO 2500
2070 IF T$=CR$ THEN H=H-1:GOTO 2500
2080 IF T$=CD$ THEN V=V+1:GOTO 2500
2100 IF T$=CU$ THEN V=V-1:GOTO 2500
2110 IF T$=RT$ THEN H=0:V=V+1:GOTO 2530
2115 IF T$=F1$ THEN POKE PO,X:GOTO 7000
2120 IF T$=F2$ THEN POKE PO,X:GOTO 7100
2125 IF T$=F3$ THEN POKE PO,X:GOTO 7400
2130 IF T$=F4$ THEN POKE PO,X:GOTO 7300
2135 IF T$=F5$ THEN POKE PO,X:GOTO 6200
2140 IF T$=F6$ THEN POKE PO,X:GOTO 6000
2145 IF T$=F7$ THEN POKE PO,X:GOTO 2600
2150 IF T$=HM$ THEN POKE PO,X:GOTO 2010
2155 IF T$="+" THEN 2200
2160 IF T$="-" THEN 2300
2165 IF T$<>CS$ THEN 2030
2170 FOR I=0 TO 12
2175 POKE FX+13*CH+I,0:NEXT
2180 GOSUB 1000:GOTO 2010
2200 REM + SERVICE
2205 IF CP=225 THEN 2400
2210 IF X=PD OR X=CX OR X=BX THEN 2280
2220 IF X=LX OR X=DX THEN 2290
2240 X=CX
2260 IF Y=LX THEN Y=BX:GOTO 2290
2270 IF Y=DX THEN Y=RX:GOTO 2290
2275 GOTO 2290
2280 X=CX
2290 H=H+1:GOTO 2500
2300 REM - SERVICE
2310 IF X=PD OR X=CX OR X=BX THEN 2350
2320 IF X=LX THEN 2290
2330 IF X=DX THEN X=LX:GOTO 2260
2340 X=BX:GOTO 2260
2350 X=BX:GOTO 2290
2400 IF X=PD OR X=BX THEN X=RX:GOTO 2410
2402 IF X=CX OR X=RX THEN X=RX:GOTO 2410
2404 IF Y=PD OR Y=BX THEN Y=LX
2406 IF Y=CX OR Y=LX THEN Y=LX
2408 X=DX:GOTO 2290
2410 IF Y=LX OR Y=RX THEN Y=DX
2415 IF Y=PD OR Y=BX THEN Y=LX:GOTO 2290
2420 IF Y=CX OR Y=LX THEN Y=LX:GOTO 2290
2450 IF Y=PD OR Y=BX THEN Y=LX:GOTO 2290
2455 IF Y=LX OR Y=RX THEN Y=DX
2499 GOTO 2290
2500 IF H<0 THEN H=10:V=V-1
2510 IF H>5 THEN H=0:V=V+1
2520 IF V<0 THEN V=5
2530 IF V>7 THEN V=00
2540 POKE PO,X:POKE(PO+1),Y
2550 GOTO 2020
2600 REM REVERSE - AS BEST YOU CAN !
2610 XX=OG
2620 FOR I=0 TO 7
2630 FOR J=0 TO 5:XJ=XX+J
2640 A=PEEK(XJ):IF A=PD THEN A=BX
2650 IF A=BX THEN A=CX:GOTO 2700
2660 IF A=CX OR A=DX THEN A=BX:GOTO 2700
2670 IF A=RX THEN 2720
2675 IF PR=BX THEN A=CX:GOTO 2700
2680 IF PEEK(XJ+1)=RX THEN A=RX:
     GOTO 2700
2690 A=BX
2700 PR=A:POKE(XJ),A:NEXT
2710 PR=0:XX=XX+RW:NEXT:GOTO 2010
2720 IF PR=DX OR PR=RX THEN A=LX:
     GOTO 2700
2730 A=BX:GOTO 2700
2800 IF PR=DX OR PR=RX THEN A=LX:
     GOTO 2700
3000 PRINT"S";QQ$;LL$;"I";:INPUT CH
3010 CH=CHAND255
3020 PRINT"S";QQ$;LL$;S9$
3030 RETURN
4000 PRINT"□";QQ$;
     "+ ON / - OFF / £ HALF / ^ FULL"
```

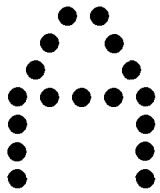
```
4010 PRINT"QF1 READ  CHAR #"
4020 PRINT"F2 WRITE CHAR #"
4030 PRINT"F3 READ NEXT"
4040 PRINT"F4 WRITE CURRENT/READ NEXT"
4050 PRINT"F5 DISK LOAD"
4060 PRINT"F6 DISK SAVE"
4070 PRINT"F7 REVERSE CHARACTER"
4080 RETURN
5000 PRINT CHR$(142)
5010 POKE 52,48:POKE 56,48
5020 POKE 56334,PEEK(56334) AND 254
5030 POKE 1,PEEK(1) AND 251
5040 FOR I=0 TO 2047
5045 POKE I+12288,PEEK(I+53248):NEXT
5050 XX=12288+91*8
5060 FOR I=0 TO 39
5065 READ A:POKE(XX+I),A:NEXT
5070 POKE 1,PEEK(1) OR 4
5080 POKE 56334,PEEK(56334) OR 1
5090 POKE 53272,PEEK(53272) AND 240 OR 12
5095 RETURN
5100 POKE 650,128:RT$=CHR$(13)
5110 OG=1024:RW=40:FX=256*56
5120 POKE 53281,1:POKE 53280,2
5130 PD=ASC("."):PRINT"■";
5140 BL=123:CL=124:LH=125:LR=126:RH=127
5150 BX=91:CX=92:LX=93:DX=94:RX=95
5160 CH=0:F1$="■":F2$="■":F3$="■"
5165 F4$="■":F5$="■":F6$="■":F7$="■"
5170 CL$="]":CR$="■":CU$="■"
5175 CD$="Q":HM$="S":CS$="■"
5180 ZL$=RT$+"■]]]]]]]]]]]]]]"
5185 ZS$="                              "
5190 ZR$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
5195 EL$=ZL$+ZS$+ZR$:EU$=CU$+EL$
5196 QQ$="QQQQQQQQQQ"
5197 LL$="]]]]]]]]]]]]]]]]]]"
5198 RETURN
6000 REM SAVE TO DISK
6010 PRINT"SQQQQQQQQQQQQQQQQQ";
6020 B$="":PRINT EL$;:INPUT "FILE";B$
6025 IF B$="" THEN PRINT CU$;:GOTO 6160
6030 C$="@0:"+B$+",S,W"
6040 OPEN 3,8,8,C$
6050 PRINT EU$;:INPUT "FROM CHAR #";FC%
6060 PRINT EU$;:INPUT "  TO CHAR #";TC%
6070 IF TC%<FC% THEN 6050
6080 PRINT#3,FC%;RT$;TC%;RT$;
6090 FOR I=FC% TO TC%
6100 PRINT#3,139;RT$;:REM ATTRIBUTE
6110 II=FX+I*13
6120 FOR J=II+1 TO II+11
6125 PRINT#3,PEEK(J);RT$;:NEXT J
6130 NEXTI:CLOSE3
6140 PRINT EU$;B$;" SAVED";:GOTO 2010
6200 REM LOAD FROM DISK
6210 PRINT"SQQQQQQQQQQQQQQQQ";
6220 B$="":PRINT EL$;:INPUT "FILE";B$
6225 IF B$="" THEN PRINT CU$;:GOTO 6320
6230 C$="@0:"+B$+",S,R"
6240 OPEN 3,8,8,C$
6250 INPUT#3,FC%,TC%
6260 FOR I=FC% TO TC%
6270 II=FX+I*13:INPUT#3,C%:POKE II,0
6280 FOR J=1 TO 11
6285 INPUT#3,C%:POKE(II+J),C%:NEXT J
6290 POKE(II+12),0:NEXT I
6295 CLOSE 3:GOTO 2010
6300 PRINT"■";EL$;B$;" LOADED";:GOTO 2010
6600 PRINT RT$;"■]]]]]]]]]]]]]]]]]]";
6602 PRINT "                              ";
6604 PRINT "■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
6610 RETURN
7000 REM READ SPECIFIED CHARACTER
7010 PRINT HM$;:GOSUB 7200
7020 IF CC%<256 THEN CH=CC%
7030 CH=CC%:GOSUB 1000:GOTO 2010
7100 REM WRITE SPECIFIED CHARACTER
7110 PRINT HM$;CD$;:GOSUB 7200
7120 IF CC%<256 THEN CH=CC%
7130 GOSUB 1400:GOTO 2010
7200 PRINT QQ$;"Q";
7210 PRINT RT$;"]]]]]]]]]]]]]]]]";
7220 PRINT CH;"■          ■■■■■■■■■■■■■";
7230 CC%=CH:INPUT CC%
7235 IF CC%<0 OR CC%>255 THEN CC%=256
7240 RETURN
7300 REM WRITE CURENT/READ NEXT
7310 GOSUB 1400
7400 REM READ NEXT
7410 CH=((CH+1) AND 255)
7420 GOSUB 1000:GOTO 2010
9000 DATA 0,0,0,0,0,0,0,0
9010 DATA 0,60,126,126,126,126,60,0
9020 DATA 0,192,224,224,224,224,192,0
9030 DATA 0,195,231,231,231,231,195,0
9040 DATA 0,3,7,7,7,7,3,0
```

is designed to provide you enough information to understand the operation of the FX80 without the manual, and, to provided actual programs that make the whole process easier. The FX80 allows the user to define any number of characters, up to 256 characters in the standard RAM area, more with additional RAM. First consider each character in the FX80 to be defined in a matrix that is eight dots high by six dots wide. This is similar to the character definition on the various microcomputers, although many of them are defined as eight by eight. The capital letter A would be defined as:



The alphanumerics and punctuation will normally be a five by seven character within the six by eight matrix, providing at least one blank column and one blank row between individual characters. Simple enough. But, the FX80 people decided to get even better looking characters

by allowing dots to be placed half way between the six columns. This means that there are not six but eleven possible horizontal positions for dots. The capital letter **A** could be defined as:



Note that the dots in the top half of the letter **A** overlap vertically. The dots on the top line and third line are offset one-half space from the dots on the second and fourth through seventh lines. This extra resolution in the horizontal direction, eleven positions in the normal space of six, provides very nice looking character, but does severely complicate the character generation process. There is an important restriction to be aware of. The dots must not overlap. If there is a dot on one of the eleven positions, then there may **not** be a dot on the next position.

The FX80 manual discusses how to define your character on graph paper, and then to manually convert each of the eleven columns into an 8-bit byte. The top most row is considered to have a value of 128 decimal or 80 hex. The next row is 64 decimal, 40 hex, and so forth. The value of each dot that is to be printed in any column is summed with all other dot values in the column to produce the column value. There is one 8-bit value for each column, and eleven columns. In addition to these eleven bytes of information to define each character, you must provide an attribute byte. The attribute byte determines if the character is to be shifted down one dot position to provide a lower case descender, which column the character starts in and which column it ends in. This is provided to permit proportional spacing. In this article, we assume that all characters are full width and print in the upper eight dot positions.

If you sit down and start plotting out characters on graph paper, then assigning each dot its value, summing the individual dot values, and so forth, that it is a somewhat tedious, repetitive job. The type of job that computers are supposed to be good at! So, why not have the computer do the work! The FX80 GENERATOR is given in **Listing 1**. It is based on the interactive Programmable Character Generator by Loren Wright that appeared in the **GRAPHICS** issue. Several important additions have been made by me. First, the program automatically defines a few new characters for use on the Commodore 64. The FX80 can printed dots between the main columns of each character; the Commodore can not. To make a realistic display of the character being generated, three new characters are defined. They are the left half of a dot, the right half of a dot, and the left and right halves joined 'back-to-back'. This permits a dot to be generated that appears to be half way between the normal columns of the Commodore display. Other enhancements include **Save to Disk**, **Load from Disk**, and **Reverse Character**.

The operation of the program is quite straight-forward. After the three special half dots are generated, the display is initialized. A character is generated in the upper left corner of the display. The + key is used to write a dot; the - key to erase a dot; the **up arrow** to select the main columns; and, the British pound sign to select the dots on the intermediate columns. The **HOME**, **CLR** and the cursor controls can be used for their normal functions. **Function Keys** 1 through 7 each have a specific function associated with them. These functions are listed on the display.

F1 **READ CHAR #** converts eleven bytes of data in memory accessable by the character number into a dot display in the upper left corner.

F2 **WRITE CHAR #** converts the dot pattern in the working space into eleven bytes of data in memory accessable by the character **F3** **READ NEXT** reads the next character without requiring a number to be specified.

F4 **WRITE CURRENT/READ NEXT** combines a write with a read to make entering of a number of consecutive characters more automatic.

F4 **DISK LOAD** loads a file specified by name from the disk to make it available for editing.

F6 **DISK SAVE** saves the character information that is in memory on the disk.

F7 **REVERSE CHARACTER** changes the dots on the main columns to blanks and the blanks to dots. Note that this will only work well with patterns that have no dots between the main columns. This is because the while the FX80 can print half spaces, it can not print half dots. Think about it. It is not as simple as it seems.

Program Description

The subroutine at **5000** copies the standard ROM character set into RAM. Five new characters are defined by reading the DATA statements at **9000** and poking them into the character RAM. These five characters are a blank, a solid dot, the left half of a dot, the left and right halves of a dot joined 'back-to-back', and the right half of a dot. They are

SOPHISTICATED TELE-COMMUNICATION IS HERE

THE COMMUNICATOR

for 4.0 Commodore Computers

JIM STRASMA'S REVIEW:
"THE BEST TERMINAL PACKAGE I'VE SEEN YET"

By April 1 (maybe sooner) It Will Be Even Better

SPEEDS UP TO 9600 BAUD
XON — XOFF
TRUE CTRL KEY (we do our own keyboard scan)

THE HARDWARE — A printed circuit board; easily installed in the CBM. It uses no CBM connectors; gives a serial port with true RS232C standard.

THE SOFTWARE —
- Emulates the ADDS Regent 100, ADM 31 and/or the TeleVideo 950.1 Or choose the VT100 model for use with DEC and VAX computers.
- Runs coresident with BASIC programs; lets BASIC programs and program on host computer communicate to develop really sophisticated communication and control capabilities.
- The program is on ROM at either address; no disk loading required. Uses only 512 bytes of RAM; will relocate itself around any other machine language program at top of memory.
- Will upload and download and run BASIC programs. With BASIC program will upload and download standard data files. 100 page manual gives program listing for BASIC programs.

Excellent text editor designed to work with THE COMMUNICATOR

THE COMMUNICATOR $200
Text Editor $40

1200 baud modems beginning at low, low $385, and even less when purchased with THE COMMUNICATOR

AMPLIFY, INC.
2325 Macbride, Iowa City, Iowa 52240 319-337-8378

1 trademarks Adds Regent, Inc., Lear Liegler, Inc., Televideo Systems, Inc.

used to provided the intermediate dots for the even columns in the working display. The **Character Dot-Data Base Address** at **53272** is set to point at the RAM character set.

The subroutine at **5100** defines a number of variables and strings.

Subroutine **4000** displays the basic editing screen.

Subroutine **1000** examines the data in memory for a specified FX80 character. Each character is represented by 13 consecutive memory locations. To make the testing easier, the first and last location are always zero. The other eleven locations contain the bit pattern for the FX80 eight-by-eleven dot matrix. Since the intermediate dots, those on the even numbered columns, are generated by a combination of display patterns - the specially defined patterns generated above - the prior, current and next memory locations all must be tested to determine the correct pattern to display.

Subroutine **1400** reverses the process and converts the display pattern into the correct memory pattern.

The major editing service starts at **2000**. It displays the current character number and sets the cursor shape. There are two cursor shapes used in this program. The first is an open circle that represents the odd columns. It is selected by the **Vertical Arrow** located next to the **RESTORE** key. When it is selected, you can write or erase in the six odd columns. The second cursor is a half solid vertical bar that represents the even columns. It is selected by the **British Pound Sign** key, and when selected allows you to write or erase in the five even columns. Cursor up, down, left and right; carriage return, home and clear work area are supported. A dot is written by positioning the cursor in the desired location, including selecting the correct full or half cursor, and pressing the + key. A dot is erased by pressing the - key.

The trickiest part of the program is probably in lines **2200** through **2550** that determine what character(s) should be displayed as a function of the current mode - odd or even column, the command to write or erase, and the information that is currently in the work area. For example, if you are going to write a dot in an even column, you must actually write a right half dot in one character position and a left half dot in the next. But what if the next position, the odd column, already contains a full dot? Or, even more difficult, what if it contains the right half for the next even column? A number of tests are made to determine how the display must be changed.

Lines *600* to **2800** are a limited **REVERSE** function. It is easy to reverse a full dot by making it a space, or reverse a space by making it a full dot. The problem comes in reversing a half dot. While the FX80 can print a half space by proper positioning of the printing mechanism, it **can not** print a half dot! The print head is only capable of printing a full dot. So, it is not possible to accurately reverse patterns that contain any half dots. The routine does the best it can. When done reversing, it accepts a new character number.

The routine at **6000** to **6140** saves the FX80 character information on disk. It is called by the **F6** key. It requests a filename, the number of the first character to be saved and the number of the last character to be saved. It outputs an arbitrary attribute byte of 139 which defines each character as the upper eight printing positions on the FX80 with the full character width. The eleven bytes of data that define the information for the eleven columns on the FX80 are then output.

The routine at **6200** to **6300** loads the FX80 character information from the disk. It is called by the **F5** key.

Lines *7000* to *7420* support the other basic editing functions that include: Read a specified character, Write a specified character, Read the next character, and, Write the current character number and Read the next.

Generating Data Statements

Now that a new character set for the FX80 has been generated using the **FX80 Character Generator**, and is saved on disk as a sequential disk file, the question arises: How do I get it over to the FX80? The obvious answer is to first convert the sequential file to BASIC DATA statements and then add a driver program that would output the character information, plus any special command information required by the FX80. Step one, generating the BASIC DATA statements from the sequential file is simply handled by the **FXPCG DATAMAKER** routine. Load and run the **FXPCG DATAMAKER**. It will ask for the **FIRST BASIC LINE NO:**. Input a line number, say 10010. It will then ask for the **FILENAME:**. Input the filename that you used to save the character information. The file will be loaded from disk.

```
 1 REM FXPCG DATAMAKER - R.M.TRIPP 1983
 2 INPUT "□FIRST BASIC LINE NO: ";LN
 3 POKE52,48:POKE56,48:C=256*56:INPUT "FILENAME: ";B$:C$="@0:"+B$+",S,R"
 4 OPEN 3,8,8,C$:PRINT"LOADING":INPUT#3,F%,T%:FOR I=0 TO T%-F%:X=C+I*12
 5 FOR J=0 TO 11:INPUT#3,C%:POKE(X+J),C%:NEXT J:NEXT I:E=X+11:CLOSE 3
 6 PRINT "□";MID$(STR$(LN),2);" DATA";F%;CHR$(44);T%:LN=LN+10:K=1:GOTO 8
 7 PRINT "□";:K=0
 8 FOR I=K TO 6:PRINT MID$(STR$(LN),2);" DATA ";:FOR J=0 TO 11
 9 PRINT MID$(STR$(PEEK(C)),2);CHR$(44);:C=C+1:IF C>E THEN PRINT 256;:I=6:J=11
10 NEXT J:PRINT CHR$(20):LN=LN+10
11 NEXT I:PRINT"LN=";LN;":C=";C;":E=";E
12 IF C>E THEN PRINT"QQGOTO 14":GOTO 16
13 PRINT "QQGOTO 7":GOTO 16
14 PRINT "□";:FORI=1TO8:PRINT I:NEXT:PRINT "GOTO 15":GOTO 16
15 PRINT "□";:FORI=9TO16:PRINT I:NEXT
16 POKE 631,19:FOR I=1 TO 9:POKE 631+I,13:NEXT:POKE 198,10:END
```

You will then see DATA statements appear on the display, the display be cleared, and more DATA statements appearing, until all of the characters have been converted to DATA statements. Then the numbers **1** to **16** will appear on the screen. At this point, the program is erasing itself! You are left with a BASIC program that is nothing but DATA statement containing your character set.

Now all that is left is to add a short BASIC program in front of these DATA statements that will send initialization information to the FX80. The **Character Set for FX80** shows the additional DATA that is needed to set up a typical FX80 application. Lines **10001** to **10009** are the sequences that must be sent to the FX80 to perform the specified operations. See the FX80 manual for additional details.

All that you need to add is a driver program that will open up your communication channel, read and output the information contained in the DATA statements, and you are on your way. The FX80 will now contain the characters you defined.

Using the New Characters

You will need to write a short conversion program to output the correct codes to the FX80 to access the characters you have generated. The **SCREEN DUMP** program is an example that was used with my character set. It can be appended to an existing screen editing program by listing it on the display, loading you editor, and then pressing returns on each line of the display. Then add a command to your editor to go to the **30000** subroutine, and you can dump your screen directly to the

```
30000 REM SCREEN DUMP
30001 OPEN 3,4
30002 FOR Y=0 TO 24
30003 FOR X=0 TO 39
30004 A=PEEK(1024+Y*40+X)
30005 IF A<32 THEN A=A+64:GOTO 30010
30006 IF A>63 AND A<128 THEN A=A+64:
      GOTO 30010
30007 IF A>127 AND A<192 THEN A=A-128:
      GOTO 30010
30010 PRINT#3,CHAR$(A)
30011 NEXT:PRINT#3,CHR$(13)
30012 NEXT:PRINT#3,CHR$(13)
30013 CLOSE 3:RETURN
```

FX80. Your conversions will vary as a function of how you have defined the special characters on the FX80.

What's Next

A scheme to intercept the output of a BASIC LIST command and convert it will be presented next month. A complete **PARALLEL** driver will be described that includes the cabling information and software to permit you to use the FX80 with your Commodore without requiring a serial interface! This can save you quite a bit of money since the serial interface on the FX80 is an option, the parallel interface is standard. While it does take a bit of effort to get the FX80 working with the special characters for your computer, it does provide you with a virtually unlimited printing capability.

MICRO™

HOW MUCH LONGER WILL YOU LAST?

How long can you endure? When will it end?... We're not talking about a new shoot 'em up game for the Commodore 64, but Commodore's own disk operating system! Commodore made a great computer in the 64 but left its disk operating system out in the cold. If you've been waiting for a true disk operating system, here it is!... If you've been waiting for a great BASIC language enhancement that will let you utilize the Commodore's many special features, here it is! What is it? It's grafDOS, the great new utility from Xylex Software that allows the user to actually become friendly with the Commodore 64! grafDOS includes commands like DELETE, RENAME, CATALOG, RUN, etc. The BASIC allows you to do high resolution and low resolution graphics, sound, sprite program, plus much, much more for a total of 40 commands! Plus included in every package is MINIMON, a powerful machine language monitor that includes another 20 commands for use in machine language. The disk also comes with sample programs and demos including a great music generator! And all this together is only $49.95! How could you have lasted this long without it?

DON'T WAIT ANY LONGER!

Make your programming easier! grafDOS is available now at your local dealer or:

INTERESTING SOFTWARE
21101 S. Harvard Blvd.
Torrance, CA 90501
(213) 328-9422

Visa/MC/Check/Money Order
Add $2.00 shipping
CA residents add 6½# sales tax.
Dealer inquiries invited.

Commodore Reviews

Product Name: **Interpod**
Equip. req'd: Commodore VIC or 64
Price: $150.00
Manufacturer: Oxford Computer Systems Ltd.
Hensington Road, Woodstock
Oxford OX7 1JR England
U.S. Distributors:

SJB Distributors
10520 Plano Rd., #110
Dallas, TX 75238
(214) 343-1328

Limbic Systems
1056 Elwell Ct.
Palo Alto, CA 94303
(415) 964-8788

Description: This is an IEEE-488 and RS-232 interface for the VIC-20 and C64. It is guaranteed compatible with all programs. Though it lacks the BASIC 4 and monitor of RTC's C64-LINK, INTERPOD is a most important new product. Highly recommended.

Pluses: Unlike other IEEE interfaces we've tested, it connects to the serial bus, like Commodore accessories. Thus, there is no need to relocate it and no problem with programs that use up internal memory. It easily links a VIC-20 or C64 to a whole network of PET/CBM or other IEEE-488 devices plus an RS-232 printer, and it works fine with most DOS-protected programs. When the computer addresses a device, *INTERPOD* sees if a device on the serial bus answers. If not, the call is transferred to the IEEE bus to see if any device answers there. If not, and the device number is 4, it tries the RS-232 port. It ignores the IEEE device when a serial device is on and can arrange another address for the RS-232 device.

Minuses: *INTERPOD* goes off and sulks if you turn the computer off and on without also turning off *INTERPOD*. A multi-plug switch solves this. When rechecking IEEE disk status, *INTERPOD* adds garbage after the valid information; this hangs some data-handling programs. The cure, says the fairly adequate manual, is to send a "clear" command to *INTERPOD's* command channel before each new access — easy in your own programs, but impossible in protected commerical ones.

Skill level required: None to use the IEEE port. The usual high skill is needed to set up the RS-232 port.

Reviewer: Jim Strasma

Product Name: **DIARY 64**
Equip. req'd: Commodore 64 Computer, 1541 Disk drive or Dattasette; Printer Optional
Price: $39.95
Manufacturer: Computer Marketing Services
300 W. Marlton Pike
Cherry Hill, NJ 08002
(609) 795-9480

Description: *DIARY 64* is a cartridge based file management program. Files are set up in blocks which may consist of up to 10 lines of 27 characters each. When the files are printed out, only the first four lines of each block is printed. This allows printing names and addresses or like information. The files may be saved on either tape or disk.

Pluses: The program is quite easy to use. The manual explains the functions quite well. Files may be set up using a date rather than a block number and the file information can be accessed by data as you would use a diary. This seems to be the origin of the program's name. Blocks may be added, changed or deleted as necessary. There is a search function allowing a particular record to be found by block number, data, or "key" character.

Minuses: The blocks cannot be sorted nor can you set up and print out all files at one time. Each file record requires the user to press the "up-arrow' and the return keys to obtain a print out.

Documentation: The eight page manual contains all the information needed to use the program effectively. It is clearly written and easy to understand.

Skill level required: Beginning level user.

Reviewer: Richard E. Devore

Commodore to Smart Modem

by John Kelty

Adapted for Commodore by Phil Daley

```
 10 REM SMART300
 20 REM WRITTEN BY JOHN R. KELTY
 30 REM LINCOLN, NEBRASKA  68505
 40 REM MODIFIED FOR C64 BY PHIL DALEY
 50 REM
 60 REM USE THIS PROGRAM TO CONNECT
 70 REM AND PROGRAM THE HAYES
 80 REM SMARTMODEM TO THE
 90 REM COMMODORE-64 COMPUTER.
100 REM
110 REM THE CONNECTOR CABLE SHOULD
120 REM CONNECT TO THE RS-232 PORT
130 REM IN THE FOLLOWIMG WAY:
140 REM HAYES PIN C-64 PIN
150 REM ---------- --------
155 REM 1          A
160 REM 2          M
170 REM 3          B & C
180 REM 5          K
190 REM 6          L
200 REM 7          N
205 REM 8          H
210 REM 20         E
215 REM 22         F
220 REM THE 300 BAUD HAYES MODEM
230 REM MAY BE USED WITH VIDEOTEX
240 REM OR ANY OTHER TERMINAL
250 REM PACKAGE THAT ALLOWS THE
260 REM HAYES TO RUN AT 300 BAUD.
270 REM
280 REM THE HAYES WILL OPERATE
290 REM LOCALLY AT HIGHER RATES
300 REM BUT NO HIGHER THAN 300
310 REM WHILE ON LINE.
320 REM
330 REM SEE YOUR USER'S MANUAL
340 REM FROM HAYES FOR DETAILS.
350 REM
360 REM ****************************
370 REM
380 REM OPEN MODEM LINE FIRST THING
390 OPEN 2,2,3,CHR$(6)
420 DIM M$(5),SN$(9)
430 REM ****************************
510 PRINT"□@    ***** SMART MODEM
    *****"
520 PRINT"      BY: KELTY ENGINEERING"
530 PRINT"
    ************************"
540 PRINT:
    PRINT"CONNECT CABLES TO MODEM,"
545 PRINT"COMPUTER AND PHONE
    COMPANY."
550 PRINT:PRINT:
    PRINT"(BREAK AND LIST THIS
    PROGRAM TO"
555 PRINT"SEE CABLE DETAILS)."
560 PRINT:PRINT:
    INPUT"PRESS ENTER WHEN READY";A$
570 PRINT"□@"
580 PRINT#2,"AT Z":'CLEAR MODEM
590 REM ****************************
600 REM MENU
610 M$(1)="AUTO DIALING"
620 M$(2)="AUTO ANSWERING"
630 FOR S=1 TO 2
640 PRINTS;M$(S)
650 NEXT S
660 PRINT:PRINT:
    PRINT"SELECT ONE OF THE ABOVE"
670 GET A$:IF A$=""THEN670
680 ON VAL(A$) GOSUB 700,1310
690 GOTO 510
700 REM ****************************
710 REM AUTO DIALING ROUTINE
720 REM ****************************
730 REM
740 PRINT"□@"M$(1)
750 PRINT:PRINT:
    PRINT"1 DIAL A SAVED NUMBER"
760 PRINT"2 DIAL A NEW NUMBER"
770 GET A$:IFA$=""THEN770
780 ONVAL(A$) GOTO800,980
790 GOTO740
800 REM ************************
810 HM=2:'HOW MANY NUMBERS
820 REM TO CHANGE THIS LIST,
    MAKE HM=HOW
825 REM MANY # YOU WANT TO LIST (MAX
    OF 9).
```

```
 830 REM THEN INSERT LINES SIMILAR TO
     THOSE
 835 REM BELOW WITH YOUR NUMBERS AND
     NAMES.
 840 PRINT"■NUMBERS YOU CAN EASILY
     CALL":PRINT
 850 SN$(1)
     ="1   XXX-XXXX          A FRIEND
     "
 860 SN$(2)
     ="2   XXX-XXX-XXXX      A BBS
     "
 870 FOR K=1 TO HM
 880 PRINT SN$(K)
 890 NEXT
 900 PRINT:
     PRINT"SELECT ONE OF THE ABOVE"
 910 PRINT"(BREAK AND LIST TO EDIT
     THESE   NUMBERS)."
 920 GET A$:IF A$=""THEN920
 930 NB=VAL(A$)
 940 IF(NB<1)OR(NB>HM)THEN 800
 950 P$=MID$(SN$(NB),5,16)
 960 GOSUB1090
 970 RETURN
 980 REM ************************
 990 PRINT:PRINT"DIALING A NEW NUMBER"
1000 PRINT:
     PRINT"TYPE IN THE PHONE NUMBER
     THAT"
1005 PRINT"YOU WISH TO CALL."
1010 PRINT"EXAMPLES:"
1020 PRINT"1-800-XXX-XXXX    LONG
     DISTANCE
1030 PRINT"112-800-XXX-XXXX  FROM
     LINCOLN
1040 PRINT"XXX-XXXX          LOCAL
     CALL
1050 PRINT
1060 INPUT"TYPE DESIRED NUMBER";P$
1070 GOSUB 1090
1080 RETURN
1090 REM ************************
1100 REM NUMBER KNOWN AT THIS POINT
1110 REM READY TO DIAL
1120 PRINT#2,"AT F0 D"P$
1160 REM ************************
1170 REM YOU CAN RUN ANY TERMINAL
1180 REM PACKAGE THAT YOU WISH BY
1190 REM EDITING THE NEXT LINES.
1200 REM LOAD"MODCOMM.PRG",8
1220 REM RUN
1230 PRINT"■@YOU NEED TO USE YOUR
     OWN"
1240 PRINT"TERMINAL PACKAGE AT THIS
     POINT"
1250 PRINT"(SUCH AS VIDEOTEX AND
     OTHERS)."
1300 GOTO 2040
1310 REM ********************
1320 REM AUTO ANSWER ROUTINE
1330 REM ********************
1340 REM
1350 PRINT"■@"M$(2)
1360 PRINT
1370 REM MENU
1380 AN$(1)="DO NOT ANSWER"
1390 AN$(2)="ANSWER IMMEDIATELY"
1400 AN$(3)="ANSWER AFTER XX RINGS"
1410 FOR SA=1 TO 3
1420 PRINTSA;AN$(SA)
1430 NEXT SA
1440 PRINT:PRINT:
     PRINT"SELECT ONE OF THE ABOVE"
1450 GET A$:IF A$=""THEN 1450
1460 PRINT"■@        "M$(2)
1470 ON VAL(A$) GOSUB 1490,1560,1720
1480 GOTO 1310
1490 REM ************************
1500 REM DO NOT ANSWER
1510 PRINT:
     PRINT"      ***"AN$(1)"***"
1520 PRINT:
     PRINT"THE SMART MODEM ANSWER
     FUNCTION"
1530 PRINT"IS DISABLED."
1540 PRINT#2,"AT S0=0"
1550 GOTO 2040
1560 REM ************************
1570 REM ANSWER IMMEDIATELY
1580 PRINT"   ***"AN$(2)"***"
1590 PRINT"THE SMART MODEM WILL
     ANSWER ALL"
1600 PRINT"CALLS IMMEDIATELY."
1610 PRINT"USE THIS TO TRANSFER CALLS
     THAT"
1620 PRINT"ARE ALREADY IN PROGRESS
     BETWEEN"
1630 PRINT"TWO INDIVIDUALS TO
     COMMUNICATION"
1640 PRINT"BETWEEN THEIR COMPUTERS."
1650 GOSUB 1870
1660 PRINT#2,"AT"DP$"A"
1670 PRINT"AT THIS POINT YOU NEED TO
     RUN"
1680 PRINT"OR EXEC YOUR OWN RECEIVER"
1690 PRINT"PROGRAM TO INTERPRET THE"
1700 PRINT"INCOMING CALL."
1710 GOTO2040
1720 REM ********************
1730 REM ANSWER AFTER XX RINGS
1740 PRINT:PRINT"  ***"AN$(3)"***"
1750 PRINT:
     PRINT"THE SMART MODEM WILL
     ANSWER ALL"
1760 PRINT"CALLS AFTER THE NUMBER OF
     RINGS"
1770 PRINT"YOU SELECT (FROM 1 TO 255
     RINGS)"
1780 PRINT
1790 INPUT"HOW MANY RINGS (1 TO 255)";
     RG
1800 RG=INT(RG)
1810 IF RG<1 OR RG>255 THEN 1790
```

```
1820 PRINT:
     PRINT"THE PHONE WILL BE
     ANSWERED"
1830 PRINT"AFTER"RG"RINGS."
1840 GOSUB 1870
1850 PRINT#2,"AT"DP$"S0="RG
1860 PRINT:GOTO 1670
1870 REM ************************
1880 REM FULL OR HALF DUPLEX
1890 PRINT:
     PRINT"DO YOU WANT FULL OR HALF
     DUPLEX?"
1900 PRINT"(IF YOU DO NOT KNOW,
     TRY HALF)."
1910 PRINT:PRINT"1    FULL DUPLEX"
1920 PRINT"2    HALF DUPLEX"
1930 GET A$:IF A$=""THEN1930
1940 ON VAL(A$) GOTO 1950,1990
1950 REM FULL DUPLEX
1960 DP$="F1"
1970 PRINT:
     PRINT"FULL DUPLEX SELECTED."
1980 GOTO 2020
1990 REM HALF DUPLEX
2000 DP$="F0"
2010 PRINT:
     PRINT"HALF DUPLEX SELECTED."
2020 PRINT
2030 RETURN
2040 REM ************************
2050 REM EXIT ROUTINE
2080 PRINT:
     PRINT"THIS PROGRAM HAS ENDED."
3000 REM THIS PROGRAM SENDS AND
     RECEIVES
3010 REM TRUE ASCII DATA
3020 DIMF%(255),T%(255)
3200 FORJ=32TO64:T%(J)=J:NEXT
3210 T%(13)=13:T%(20)=8:RV=18:CT=0
3220 FORJ=65TO90:K=J+32:T%(J)=K:NEXT
3230 FORJ=91TO95:T%(J)=J:NEXT
3240 FOR J=193 TO 218:K=J-128:T%(J)=K:
     NEXT
3250 T%(146)=16:T%(133)=16
3260 FOR J=0 TO 255
3270 K=T%(J)
3280 IF K<>0THEN F%(K)=J:F%(K+128)=J
3290 NEXT
3300 PRINT " "CHR$(147)
3310 GET#2,A$
3320 IF A$=""OR ST<>0 THEN 3360
3330 PRINT " "CHR$(157):
     CHR$(F%(ASC(A$)));
3340 IF F%(ASC(A$))=34 THEN POKE212,0
3350 GOTO 3310
3360 PRINTCHR$(RV)" "CHR$(157):
     CHR$(146);:GET A$
3370 IF A$<>""THEN PRINT#2,
     CHR$(T%(ASC(A$)));
3380 CT=CT+1
3390 IF CT=8 THEN CT=0:RV=164-RV
3400 GOTO 3310
```

MICRO™

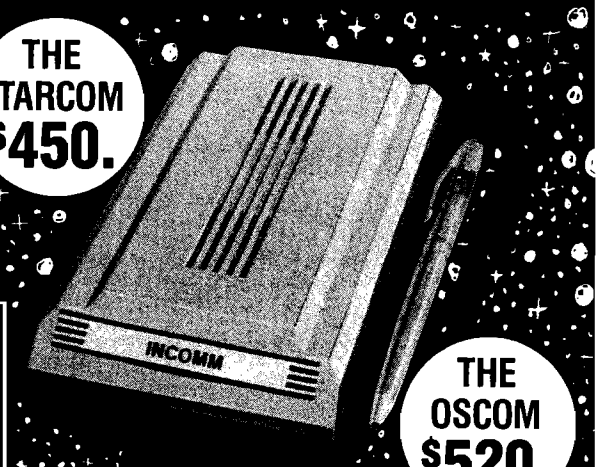
NOT ONLY ANOTHER TECHNOLOGICAL BREAKTHROUGH

BUT ALSO EASY TO USE AND TO INSTALL

Modems are the most important device used to link two products together through ordinary telephone lines at miles apart. The INCOMM STARCOM, a 300/1200 bps Auto Dial, Auto Answer, Auto Log On modem was introduced to link two high speed devices together with having absolutely NO knowledge of computers or communications in general. Anybody can install and operate the STARCOM Family in a few minutes.

THE STARCOM $450.

- STARCOM is a 300/1200 bps Auto Dial/Auto Log On/Auto Answer Modem
- OSCOM is a 300/1200 bps Auto Dial/Auto Log On/Auto Answer with Osborne (TM) Computer Software (included) modem.
- COMSOFT is a communication software package.
- OEM MODEM BOARD is also available for custom installation.



THE OSCOM $520.

SEND FOR FULL LINE CATALOG of peripherals including Modems, Muxes, Switches and Breaker Boxes

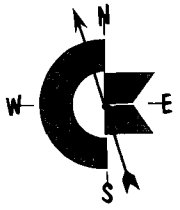
DEALER INQUIRIES INVITED
(312) 459-8881
or 1-800-323-2666

INCOMM
115 N. WOLF RD. WHEELING, IL 60090

SO COMPACT:
Only 1 1/2" x 6 1/2" x 7 1/2" 1.5 lbs.
Fits in your coat pocket!

MICRO™

Commodore Compass



by Loren Wright

High Resolution on the Commodore 64

Many C-64 users don't even know that their machines have high-resolution graphics, let alone know how to use them! There is no mention in the User's Guide, and very little in the Programmer's Reference Guide. In this column I'll explain a little about these modes, and provide a couple machine-language utilities to make high-resolution programming easier.

There are actually two bit-map modes. One is a two-color mode (hereafter called *hi-res* bit-map mode) that provides resolution of 320 by 200 pixels. The other is a four-color mode (hereafter called *multicolor* bit-map mode) providing resolution of 160 by 200 pixels. Each mode uses an 8K bit-map area and the 1K color memory.

In hi-res bit-map mode, each bit in bit-map memory corresponds to a pixel on the screen. If the bit is a 1, then the corresponding pixel appears in the foreground color; if the bit is a 0, then the corresponding pixel appears in the background color. Screen memory is used to select the foreground and background colors for each 8-pixel by 8-pixel area (the area normally occupied by a character). Bit-map memory is arranged to take advantage of this square-by-square color selection capability. The first eight bytes in bit-map memory represent the pixels of an 8-by-8 square at the top left corner. The next eight bytes of represent the 8-by-8 square immediately to the right of that. This arrangement makes it a little more difficult to calculate a given bit, but the reward is being able to use many more than two colors on a high-resolution screen.

In multicolor bit-map mode, the pixels are twice as wide and it takes two bits to specify the source of the color 00 — background #0, 01 — high four bits of screen memory location, 10 — low four bits of screen memory location, 11 — color memory location. Each byte contains four bit-pairs, each of which corresponds to a double-width pixel. The bytes are arranged in the same way as in hi-res bit-map mode, and the little squares are the same size (8 rows of 4 double-width pixels). Each little square may have only four colors in it, but there may be many more than four colors on the whole screen at once.

Listing 1 is a hi-res bit-map routine to calculate the proper byte BY, byte contents BQ, and bit BI, as well as the proper position PO for control of colors. X and Y are the input and represent the x- and y-coordinates of the hi-res pixel. 0,0 is the coordinate of the upper-left corner. BA is the starting address of bit-map memory, and OG is the screen memory start address. These vary with the memory configuration you choose.

Listing 2 is a similar routine, only for multicolor bit-map mode. X and Y are the input. Remember X ranges from 0 to 159 because of the double-width pixels. The output is the byte BY, byte contents BQ, bit pair BP, and position PO.

To use listing 1, perform the following set-up once at the beginning of your program:

```
40 DIM P%(7),M%(7):FORI=0TO7: P%(I)=2^I:
   M%=255-P%(I): NEXTI
```

The following calling sequence may now be used:

```
110GOSUB 4000: REM X=0 TO 319, Y=0 TO 199
120POKE BY,BQ AND M%(BI) OR P%(BI)
```

To use listing 2, perform the following set-up once at the beginning of your program:

```
40 DIM P%(3),M%(3):FORI=0TO3: P%(I)=4^I:
   M%=255-P%(I):NEXTI
```

You may now use the following calling sequence:

```
110GOSUB 4000: REM X=0 TO 159, Y=0 TO 199
120POKE BY,BQ AND M%(BP) OR P%(BP)
```

How to Set Up Bit Map Modes

The procedure shown in the Programmer's Reference Guide uses bit-map memory at $2000 (8192). This is adequate for experimenting with short programs, but the 6K left for BASIC is not adequate for a program of any significance. Listing 3 is a routine to set up bit-map memory at $6000 (24576). This uses bank 1 for the VIC II, so the screen must be moved to $5C00 (23552). The result is over 20K for your BASIC program. Listing 4 is a routine to restore the multicolor bit-map mode. Listing 5 is a routine to restore the normal screen. If you are using either listing 3 or listing 4, the first lines of your program must be:

```
10 POKE 52,92: POKE 56,92: POKE 55,0: CLR
20 BA=24576: OG=23552: MY=199: MX=319:
   REM MX=159 for multicolor
```

Line 10 prevents the BASIC program from writing over bit-map or screen memory. It must be the very first line in your program. Because the CLR instruction clears the stack, as well as the variables, this statement cannot be in-

cluded in a subroutine. Line 20 sets up constants used in the subroutines of listings 1 and 2.

Machine-Language Bit-Map Utilities

Listing 6 contains a group of machine-language utilities in the form of BASIC DATA statements. Space does not permit listing it in source format. Also included in the listing is a routine to POKE the utilities into the cassette buffer and subroutines to call each routine.

Important note: The routines must be POKEd back into the cassette buffer after using either the cassette or disk or LOAD or SAVE operations.

Set-up for routines:	
POKE 842,92	High byte of the beginning of screen memory.
POKE 843,96	High byte of the beginning of bit-map memory.
Clear bit map:	
POKE 840,CC	CC is used to fill bit-map memory, 0 clears to background color.
SYS 845	
Clear color memory:	
POKE 840,C	C is color to fill color memory.
SYS 883	Execute routine.
Clear Multicolor Color 1 or Hi-res Foreground	
POKE 840,C	Color used to fill high 4 bits of screen memory without disturbing low 4 bits.
SYS 898	Execute routine.
Clear Multicolor Color 2 or Hi-res Background	
POKE 840,C	Color used to fill low 4 bits of screen memory without disturbing high 4 bits.
SYS 915	Execute routine.

Miscellaneous Tidbits

If your program should happen to stop while in one of the bit-map modes, it may appear that you have lost control of your computer. One way to get out is to press STOP and RESTORE together. Another is to type GOSUB 8100 and press RETURN. Even though you can't see any characters on the screen, the command will be accepted. Keep in mind that there is only one location for color memory and it applies to whatever screen is in use at the moment. Therefore, if you are in multicolor bit-map mode and you switch to a normal character screen, then any color changes you make here will also affect Color 3 on the bit-map screen. One way to avoid this is to keep Color 3 the same for the entire screen. Another is to copy color memory to a 1K buffer at $5800. This area must be protected from BASIC by POKEing 88 instead of 92 in line 10. When you return to the bit-map screen, copy the contents of the buffer back into color memory.

Terry Peterson's BSAVE procedure (September '83 MICRO) may be used to SAVE bit-map and screen memory to tape or disk. To load, LOAD"name", dv,1, where dv is the device number. To SAVE color memory to tape, you must first copy it to a location below $8000 (32768), such as a buffer at $5800.

Finally, SAVE your program often. There is a lot to keep track of, and therefore a lot that can go wrong.

MICRO

List 1: **Calculate byte and bit in hi-res bit-map mode.**

```
4000 REM CALCULATE BYTE & PIXEL
4010 RW=INT(Y/8):CO=INT(X/8):LN=
     YAND7:BT=7-(XAND7)
4020 PO=40*RW+CO
4030 BY=BA+RW*320+CO*8+LN
4040 BQ=PEEK(BY)
4050 RETURN
```

List 2: **Calculate byte and bit pair in multi-color bit-map mode.**

```
4000 REM CALCULATE NEW PIXEL DATA
4010 RW=INT(Y/8): CO=INT(X/4)
4020 LN=YAND7: BP=3-(XAND3)
4030 BY=BA+RW*320+CO*8+LN
4040 BQ=PEEK(BY)
4050 PO=40*RW+CO
4060 RETURN
```

List 3: **Set up hi-res bit-map mode.**

```
8000REMSET UP HI-RES IN BANK 1
8001REMHI-RES SCREEN $6000-$7FFF
8002REMSCREEN MEMORY $5C00-$5FFF
8003REMMCM OFF & HI-RES ON
8010POKE56578,PEEK(56578)OR3
8020POKE56576,PEEK(56576)AND252OR2
8030POKE53272,PEEK(53272)AND7OR120
8040POKE53265,PEEK(53265)OR32
8060RETURN
```

List 4: **Set up multi-color bit-map mode.**

```
8000REMSET UP HI-RES IN BANK 1
8001REMHI-RES SCREEN $6000-$7FFF
8002REMSCREEN MEMORY $5C00-$5FFF
8003REMMCM & HI-RES ON
8010POKE56578,PEEK(56578)OR3
8020POKE56576,PEEK(56576)AND252OR2
8030POKE53272,PEEK(53272)AND7OR120
8040POKE53265,PEEK(53265)OR32
8050POKE53270,PEEK(53270)OR16
8060RETURN
```

List 5: Restore normal screen.

```
8100 REM RESTORE NORMAL SCREEN
8110 POKE 56578,PEEK(56578)OR3
8120 POKE 56576,PEEK(56576)AND252OR3
8130 POKE 53272,PEEK(53272)AND7OR16
8140 POKE 53265,PEEK(53265)AND223
8150 POKE 53270,PEEK(53270)AND239
8160 RETURN
```

List 6: Machine-language routines, installation and calling routines.

```
30 GOSUB8500:POKE843,INT(BA/256):POKE842,INT(OG/256)
3100 POKE840,C:SYS898:RETURN:REM CLEAR COLOR 1
3200 POKE840,C:SYS915:RETURN:REM CLEAR COLOR 2
3300 POKE840,C:SYS883:RETURN:REM CLEAR COLOR 3
8500 REM READ IN ML ROUTINE
8510 XX=840:I=0
8520 READYY:IFYY=-1THEN8540
8530 POKEXX+I,YY:I=I+1:GOTO8520
8540 RETURN
19000 DATA2,240,92,32,95,173,75,3,133,254,24,105,32,141,76,3,169,0,133
19010 DATA253,173,72,3,160,0,145,253,200,208,251,165,254,24,105,1,
        133,254,205
19020 DATA76,3,208,236,96,169,216,133,254,169,0,133,253,169,220,141,
        76,3
19030 DATA208,218,173,72,3,10,10,10,10,141,72,3,169,15,141,73,3,208,5
19040 DATA169,240,141,73,3,173,74,3,133,254,24,105,3,141,76,3,169,0,133
19050 DATA253,168,177,253,45,73,3,13,72,3,145,253,200,240,13,192,232,
        208
19060 DATA239,165,254,205,76,3,208,232,240,4,230,254,208,226,96,-1
```

MICRO™

FOR APPLE II PLUS, FRANKLIN, APPLE IIe

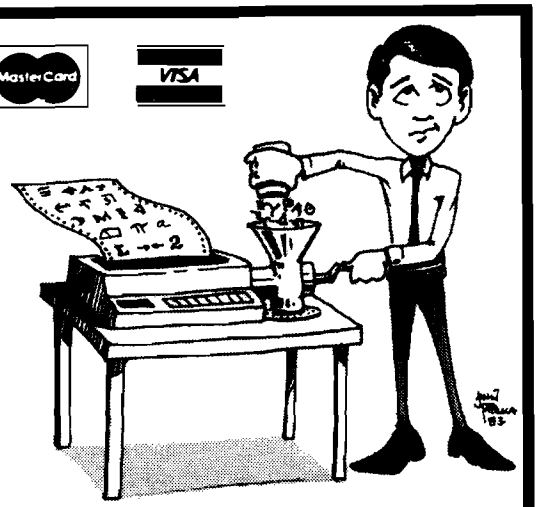
Font Down Loader

Expand the capacity of your printer hundreds of times

Load custom fonts into your Apple® Matrix Printer, Prowriter™ 8510A, OKI® Microline 92, 93, 84 Step II, and Epson® FX and use them with virtually every word processor to turn your printer into a **custom typesetter.** After the fonts are loaded, they will stay in your printer until it's turned off. A font editor is also provided to allow you to create your own graphics, text, foreign language letters, math and electronics symbols to load into your printer. On-Disk (Specify Printer)

..... $39.00

New improved versions with drivers for Grappler, Pkaso, Wizard and most other intelligent parallel boards.



"There's got to be a better way to load fonts!"

$100 REWARD

Submit the best or most unique font using the above software and we will make you $100 richer. Other prizes for the first 25 runners up.

MICRO WARE
P.O. Box 113
Pompton Plains, N.J.
07444

Dealer and Distributor Inquiries Invited

CALL (201) 838-9027

Atari To Smartmodem

By John Kelty
Adapted for Atari by Phil Daley

This is an adaptation of my program for the Color Computer. The hardware requirements are the Hayes Smart Modem and the Atari 850 commmunications interface. This program was specifically written with the serial line connected to port 1 of the serial port (the standard port for communications). This fantastic modem can be programmed from Atari BASIC.

The interface cable is a standard Atari communications cable — 9-pin D-connector to DB-25. If you want to wire your own cable, here are the connections:

Atari Connector	DB-25
1	20
2	8
3	2
4	3
5	7
6	6
7	4
8	5
9	11

You need 9-conductor cable and I have used as long as about 25 feet without any problems. It is convenient however, to have the Smartmodem near your computer setup, then you can hear the dialup and the send and receive carrier tones when you begin communicating. There are many programming options you can choose with the Hayes Smartmodem, and these are extremely well documented in the excellent Owner's Manual. (In case you have not guessed, I consider the Hayes Smartmodem my best computing purchase to date!).

The BASIC program is menu driven for ease of operation. If you BREAK and then RUN the program again, the Smartmodem functions are cleared ("AT Z" is sent to the modem). To write to the Hayes, you must access the port in concurrent I/O mode (XIO 40) and then use normal PRINT# commands. Now you can talk to your modem just like it was a printer with PRINT#5, "your modem command". To get back into the normal Atari mode use CLOSE#5. It is important to read back the information sent by the Hayes modem or else the program will hang, and pressing reset is the only way to recover. Also, the only way to terminate the program is by pressing reset.

Now you can use a communications

package such as AMODEM or a similar program to communicate with bulletin boards or data bases. The program includes a short subroutine to emulate a dumb terminal if you don't have a communications package yet.

I know little to nothing of amateur radio operations, but with a few lines of BASIC and possibly some hardware, the Hayes Smartmodem can be made to send morse code. Auxiliary relay contacts are provided in the modem to key the transmitter automatically. It cannot however, be made to directly receive morse code. Diagrams describing the modem to radio interface are found in the Smartmodem User Manual Appendix H. Obviously some nice commercial terminal packages exist for the Atari, but this program has made me happy for some time now. With the Hayes Smartmodem and the Atari Computer, I am sure you will also enjoy computer communications.

Listing 1

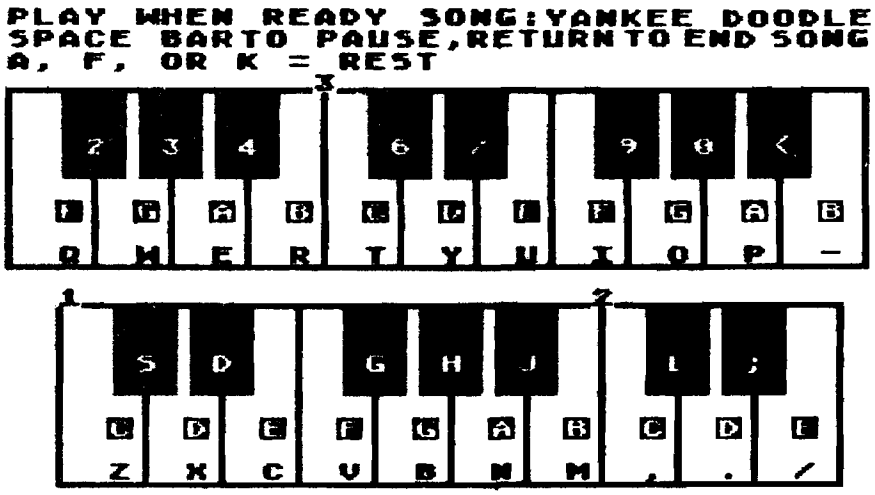
```
  5 GOTO 500
 10 REM SIMPLE TERMINAL PROGRAM
 11 REM TO GET AND PUT CHARS
 12 REM TO AND FROM MODEM
 13 OPEN #1,4,0,"K:"
 15 OPEN #3,8,0,"E:"
 20 GOSUB 400
 30 IF PEEK(764)=255 THEN 60
 40 GET #1,A:PUT #3,A
 50 PUT #5,A
 60 STATUS #5,XX
 70 IF PEEK(747)=0 THEN 30
 80 GET #5,A:IF A=0 THEN 30
 90 PUT #3,A:GOTO 30
101 REM SMART300
102 REM KELTY ENGINEERING
103 REM WRITTEN BY JOHN R. KELTY
104 REM LINCOLN, NEBRASKA  68505
105 REM
106 REM USE THIS PROGRAM TO CONNECT
107 REM AND PROGRAM THE HAYES
108 REM SMARTMODEM TO THE
109 REM ATARI COMPUTER.
110 REM
111 REM THE MODEM CABLE SHOULD
120 REM CONNECT TO THE 850
130 REM INTERFACE MODULE,
140 REM PORT #1
210 REM
220 REM THE 300 BAUD HAYES MODEM
230 REM MAY BE USED WITH AMODEM
240 REM OR ANY OTHER TERMINAL
250 REM PACKAGE THAT ALLOWS THE
260 REM HAYES TO RUN AT 300 BAUD.
270 REM
280 REM THE HAYES WILL OPERATE
290 REM LOCALLY AT HIGHER RATES
300 REM BUT NO HIGHER THAN 300
310 REM WHILE ON LINE.
320 REM
330 REM SEE YOUR USER'S MANUAL
340 REM FROM HAYES FOR DETAILS.
350 REM
360 REM ****************************
370 REM
400 OPEN #5,13,0,"R:"
410 XIO 38,#5,0,32,"R:"
420 XIO 40,#5,0,0,"R:"
430 RETURN
450 GET #5,XX:GET #5,XX
460 GET #5,XX:
    IF XX<>10 THEN PRINT CHR$(XX);
470 IF XX<>10 THEN 460
490 CLOSE #5:RETURN
500 DIM A$(30),P$(16),DP$(2)
510 PRINT "}     ***** SMART MODEM
    *****"
520 PRINT "      BY:
    KELTY ENGINEERING"
530 PRINT "
    ************************"
540 PRINT :
    PRINT "CONNECT CABLES TO MODEM,
    "
545 PRINT "COMPUTER AND PHONE
    COMPANY."
560 PRINT "PRESS ENTER WHEN READY";:
    INPUT A$
570 PRINT "}"
580 GOSUB 400:PRINT #5,"AT Z":
    GOSUB 450
590 REM ****************************
600 REM MENU
610 PRINT :PRINT "1 AUTO DIALING"
620 PRINT "2 AUTO ANSWERING"
660 PRINT "SELECT ONE OF THE ABOVE"
670 INPUT A$
680 ON VAL(A$) GOSUB 700,1310
690 GOTO 510
700 REM ****************************
710 REM AUTO DIALING ROUTINE
720 REM ****************************
730 REM
740 PRINT "}AUTO DIALING"
750 PRINT "1 DIAL A SAVED NUMBER"
760 PRINT "2 DIAL A NEW NUMBER"
770 INPUT A$
780 ON VAL(A$) GOTO 800,980
790 GOTO 740
800 REM ************************
810 HM=2:REM HOW MANY NUMBERS
820 REM TO CHANGE THIS LIST,
    MAKE HM=HOW
825 REM MANY # YOU WANT TO LIST (MAX
    OF 9).
830 REM THEN INSERT LINES SIMILAR TO
    THOSE
835 REM BELOW WITH YOUR NUMBERS AND
    NAMES.
840 PRINT "}NUMBERS YOU CAN EASILY
    CALL":RESTORE
850 DATA "1   XXX-XXXX          A
    FRIEND "
860 DATA "2   XXX-XXX-XXXX      A BBS
    "
870 FOR K=1 TO HM
880 READ A$:PRINT A$
890 NEXT K
900 PRINT "SELECT ONE OF THE ABOVE"
910 PRINT "(BREAK AND LIST TO EDIT
    THESE NUMBERS)"
920 INPUT A$:RESTORE
930 NB=VAL(A$)
940 IF NB<1 OR NB>HM THEN 800
950 FOR K=1 TO NB:READ A$:NEXT K
955 P$=A$(5,16)
960 GOSUB 1090
```

```
 970 RETURN
 980 REM ***************************
 990 PRINT "DIALING A NEW NUMBER"
1000 PRINT "TYPE IN THE PHONE NUMBER
     THAT"
1005 PRINT "YOU WISH TO CALL."
1010 PRINT "EXAMPLES:"
1020 PRINT "1-800-XXX-XXXX    LONG
     DISTANCE"
1030 PRINT "112-800-XXX-XXXX  FROM
     LINCOLN"
1040 PRINT "XXX-XXXX          LOCAL
     CALL"
1060 PRINT "TYPE DESIRED NUMBER ";:
     INPUT P$
1070 GOSUB 1090
1080 RETURN
1090 REM ***************************
1100 REM NUMBER KNOWN AT THIS POINT
1110 REM READY TO DIAL
1120 GOSUB 400:
     PRINT #5,"AT P0 T D";P$:
     GOSUB 450
1130 REM IF YOU NEED PULSE DIALING
     THEN
1140 REM SUBSTITUTE A 'P' IN ABOVE
     LINE
1160 REM ***************************
1170 REM YOU CAN LOAD ANY TERMINAL
1180 REM PACKAGE THAT YOU WISH BY
1190 REM EDITING THE NEXT LINES.
1200 REM RUN "D:AMODEM"
1230 PRINT "YOU NEED TO USE YOUR
     OWN"
1240 PRINT "TERMINAL PACKAGE AT THIS
     POINT"
1250 PRINT "(SUCH AS TELELINK AND
     OTHERS)."
1300 GOTO 2040
1310 REM ********************
1320 REM AUTO ANSWER ROUTINE
1330 REM ********************
1340 REM
1350 PRINT ")AUTO ANSWERING"
1360 PRINT
1370 REM MENU
1380 PRINT "1 DO NOT ANSWER"
1390 PRINT "2 ANSWER IMMEDIATELY"
1400 PRINT "3 ANSWER AFTER XX RINGS"
1440 PRINT "SELECT ONE OF THE ABOVE"
1450 INPUT A$
1460 PRINT ")         AUTO ANSWERING"
1470 ON VAL(A$) GOSUB 1490,1560,1720
1480 GOTO 1310
1490 REM ***************************
1500 REM DO NOT ANSWER
1510 PRINT "      ***DO NOT
     ANSWER***"
1520 PRINT "THE SMART MODEM ANSWER
     FUNCTION"
1530 PRINT "IS DISABLED."
1540 GOSUB 400:PRINT #5,"AT S0=0":
     GOSUB 450
1550 GOTO 2040
1560 REM ***************************
1570 REM ANSWER IMMEDIATELY
1580 PRINT "   ***ANSWER
     IMMEDIATELY***"
1590 PRINT "THE SMART MODEM WILL
     ANSWER ALL"
1600 PRINT "CALLS IMMEDIATELY."
1610 PRINT "USE THIS TO TRANSFER
     CALLS THAT"
1620 PRINT "ARE ALREADY IN PROGRESS
     BETWEEN"
1630 PRINT "TWO INDIVIDUALS TO
     COMMUNICATION"
1640 PRINT "BETWEEN THEIR
     COMPUTERS."
1650 GOSUB 1870
1660 GOSUB 400:PRINT #5,"AT";DP$;"A":
     GOSUB 450
1670 PRINT "AT THIS POINT YOU NEED
     TO RUN"
1680 PRINT "YOUR OWN RECEIVER"
1690 PRINT "PROGRAM TO INTERPRET
     THE"
1700 PRINT "INCOMING CALL."
1710 GOTO 2040
1720 REM ********************
1730 REM ANSWER AFTER XX RINGS
1740 PRINT "  ***ANSWER AFTER XX
     RINGS***"
1750 PRINT :
     PRINT "THE SMART MODEM WILL
     ANSWER ALL"
1760 PRINT "CALLS AFTER THE NUMBER
     OF RINGS"
1770 PRINT "YOU SELECT (FROM 1 TO
     255 RINGS)"
1780 PRINT
1790 PRINT "HOW MANY RINGS (1 TO 255)
     ";:INPUT RG
1800 RG=INT(RG)
1810 IF RG<1 OR RG>255 THEN 1790
1820 PRINT :
     PRINT "THE PHONE WILL BE
     ANSWERED"
1830 PRINT "AFTER";RG;"RINGS."
1840 GOSUB 1870
1850 GOSUB 400:
     PRINT #5,"AT";DP$;"S0=";RG:
     GOSUB 450
1860 PRINT :GOTO 1670
1870 REM ***************************
1880 REM FULL OR HALF DUPLEX
1890 PRINT :
     PRINT "DO YOU WANT FULL OR HALF
     DUPLEX?"
1900 PRINT "(IF YOU DO NOT KNOW,
     TRY HALF)."
1910 PRINT :PRINT "1    FULL DUPLEX"
1920 PRINT "2    HALF DUPLEX"
1930 INPUT A$
1940 ON VAL(A$) GOTO 1950,1990
1950 REM FULL DUPLEX
1960 DP$="F1"
1970 PRINT :
     PRINT "FULL DUPLEX SELECTED."
1980 GOTO 2020
1990 REM HALF DUPLEX
2000 DP$="F0"
2010 PRINT :
     PRINT "HALF DUPLEX SELECTED."
2020 PRINT
2030 RETURN
2040 REM ***************************
2050 REM EXIT ROUTINE
2070 PRINT "THIS PROGRAM HAS ENDED."
2080 GOTO 10
```

MICRO™

Atari Music Player

By Tom Marshall



Atari Player enables you to play music on your Atari. It converts the Atari into a simulated organ with the organ keys being represented by the keys on the Atari keyboard, as shown above. Each note you play can be heard over one of the Atari voices. The keyboard spans three complete octaves, beginning with a low C and ending more than two octaves higher with a B. Each note that is played is stored in memory so that it can be instantaneously replayed and then saved on cassette tape or disk. Later you can load your song in again and replay it.

We have included a feature in Atari Player that allows you to stop playing, go back to correct mistakes, replay the song from the beginning to the current note, and then continue playing additional notes. Absolute perfection is possible using the editing option [number 5], which allows you complete control over every note that has been played. You can change the tone, octave, and/or duration of any of the notes; or, if you wish, you can add more notes or remove them altogether.

With the Atari Player installed in your computer you can make your Atari a musically instructive as well as entertaining device.

Operating Instructions for Atari Player Demonstration

The first time you use the Atari Player, you should listen to the song already provided on your cassette or disk to appreciate how much can be accomplished. To do this, follow the simple steps listed here.

1. Using standard Atari loading procedures, load Atari PLAYER. If you are using a cassette, leave it in the cassette unit; do **not** rewind it.
2. Type 'RUN' followed by the 'RETURN' key.
3. You will be presented with the information that appears on the display screen below.
4. Press the 4 key to select the 'LOAD SONG' option.
5. The display will ask for the file name under which the song is stored.
6. If you are using a cassette, type 'C:' followed by the 'RETURN' key. This will cause the next file [a prerecorded song] on the cassette to be loaded. If you are using a disk, then type 'D:SONG' followed by the 'RETURN'. This will cause the song already provided in the file 'D:SONG' to be loaded into your Atari.
7. When the song has been loaded, Atari Player will tell you how many notes have been loaded, wait a few seconds, and then return to the original menu.
8. Press the 2 key to select the 'REPLAY SONG' option.
9. You will now be treated to a rousing song played automatically on your Atari computer.

Using the Atari Player

After choosing the 'PLAY SONG' option from the menu, the program will prompt you for the name of your song. [This is not the same thing as the file name that disk users have to supply when performing a save or a load. The name is saved with the song just for convenience and has no bearing on the song itself.] Then a representation of the Atari keyboard is printed on the screen in the format of a two-keyboard organ. The bottom row of keys represents the lowest notes, starting with 'C' and ascending alphabetically. The second row of keys represents the sharps and flats [black keys] that correspond to the first row. Note that there are inbetween keys on the second row of the Atari for every pair of first-row notes. This is different than the normal organ keyboard and means that some of the second-row keys do not sound when pressed [A, F, and K]. These keys can be used to introduce rests into your song.

The third row of keys represents the second keyboard of the organ, starting at middle 'F' and ascending to high 'B'. The top row of keys represents the sharps and flats corresponding to this second keyboard. When you have mastered the keyboard, you are well on your way to composing your own music. Read on.

At the beginning, the program waits for you to start the song. This is one of the few times when a pause doesn't count. Once you start playing, the computer keeps track of every note and its length exactly as you play it. Practice a bit to get the feel of the keyboard. It is not as simple as a piano, especially with the letters on the keys distracting you from what the true note is. The white keys on the display have the actual name of the note printed over the keyboard name of the key to help keep you oriented.

After you start a song, you may

discover that you didn't mean to play a particular note. Fortunately there is a mistake-recovery method. As soon as you realize that you have made an error (sometimes the first note is an error), press the space bar to pause momentarily. You will be presented with several options.

1. CONTINUE allows you to start playing the song at exactly the point where you stopped. This is a useful technique for the times when you become confused as to which note you want to play next; press the space bar to pause, regather your wits, and press 'C' to continue from where you stopped.
2. REPLAY will play the song up through the current note so that you can inspect your masterpiece as you input and make corrections if necessary. This option can be chosen as many times as you need it.
3. BACKUP is the option for which you've been waiting. This allows you to remove one note at a time from the current song, all the way back to the beginning if you want. When you make a mistake and press the space bar to pause, press the 'B' option and the note you are erasing will sound. Another 'B' will erase the next note, and so on. Then pressing 'C' will allow you to continue your song from the point to which you have backed up.

If, no matter how hard you try, you can't seem to get the song perfect, then the next step is to use the editor (option 5), which will allow you to manipulate the more obscure aspects of your song.

Using the Editor

The Editor option enables you to fix any minor (or major) mistakes that may creep into your performance. It allows complete control over every note in your song; the note, octave, and duration can all be changed to your specifications. And when using the insertion feature, you can start from scratch and construct your own song without ever playing a note on the Player keyboard! The Editor option is the perfect complement to Atari Player.

The first thing that the editor does is to list the notes currently in memory in groups of 30. These are notes that have been either entered from the keyboard or loaded in from a previously saved song. The 'Next' and 'Previous' commands allow you to page forward and backward through the note tables

Listing 1: **Atari Player/Editor**

```
10 REM ATARI PLAYER/EDITOR
30 OPEN #1,4,0,"K:":GOSUB 12000
40 GOSUB 10000
50 GET #1,A:IF A<49 OR A>55 THEN 5
   0
60 A=A-48:GOSUB A*1000:GOTO 40
1000 PRINT :PRINT "Input song name
   ...";:INPUT B$:IF B$="" THEN B
   $="NONAME"
1010 V=0
1015 GOSUB 14000
1020 D=0
1030 D=D+1:A=PEEK(764):IF A=255 TH
   EN 1030
1040 POKE 764,255:GOSUB 9500:SOUND
    0,0,0,0
1060 IF A=12 THEN SOUND 0,0,0,0:MA
   X=V+1:W(V+1)=0:GOTO 1120
1070 IF A=33 THEN MAX=V+1:GOSUB 80
   00:GOTO 1015
1080 IF A>62 THEN A=3
1090 TP=A:SOUND 0,ASC(PITCH$(TP+1,
   TP+1)),D$,L
1100 V=V+1:IF V=LN THEN W(V)=0:GOT
   O 1120
1110 GOTO 1020
1120 POKE CF,0:RETURN
2000 ? "PLAYING":V=1
2010 IF W(V)=0 THEN ? "?NO SONG IN
    MEMORY":GOTO 9000
2020 GOSUB 9200:IF TP=0 AND D=0 TH
   EN RETURN
2030 SOUND 0,0,0,0:GOSUB 9800:V=V+
   1:IF V>=MAX THEN SOUND 0,0,0,0
   :V=V:V=V-1:RETURN
2040 IF PEEK(764)<>255 THEN POKE 7
   64,255:RETURN
2050 GOTO 2020
3000 GOSUB 9100:OPEN #2,8,0,F$:TRA
   P 3500
3010 PRINT #2;B$
3020 FOR V=1 TO MAX:? #2;W(V):NEXT
    V
3040 CLOSE #2:? "SAVED ";MAX-1;" N
   OTES TO ";F$:GOTO 9000
3500 CLOSE #2:? "FILE I/O ERROR":G
   OTO 9110
4000 GOSUB 9100:OPEN #2,4,0,F$:TRA
   P 3500
4010 INPUT #2,B$
4020 INPUT #2;A:W(V)=A
4030 IF A>0 THEN V=V+1:GOTO 4020
4040 MAX=V:CLOSE #2:? "LOADED ";MA
   X-1;" NOTES FROM ";F$:GOTO 900
   0
5000 POKE 201,5:POKE CF,1:POKE 82,
   0
5020 V=0
5030 POKE CF,1:POKE 703,24:PRINT "
   CURRENT SONG: ";B$
5032 POSITION 0,1:PRINT "  B     N
     O   L     B     N   O   L
   ";:COLOR 2:PLOT 18,2:DRAWTO 18
   ,19
5040 FOR XX=2 TO 22 STEP 20:FOR YY
   =3 TO 17:V=V+1
5060 GOSUB 9200:IF W(V)=0 THEN MAX
   =V:POSITION XX,YY:PRINT "END":
   GOTO 5130
```

```
5062 NO=ASC(NTOCT$(TP+1,TP+1)):O=I
     NT(NO/13):Y=NO-INT(NO/13)*13:N
     $=NT$(Y*2+1,Y*2+2)
5080 POSITION XX,YY:PRINT V,N$;"
     ";O;"    ";D
5090 NEXT YY:NEXT XX
5130 POSITION 0,20:POKE 703,4:POKE
     CF,0
5140 PRINT "[N]EXT,  [P]REVIOUS, [C]URRE
     NT, OR [#]":A$="":INPUT A$:PP=V:
     IF A$="" THEN A$="Z"
5150 V=0:IF ASC(A$)>=48 AND ASC(A$
     )<=57 THEN V=VAL(A$):PP=PP-30
5160 IF V=0 THEN 5430
5170 PRINT "[C]HANGE,  [D]ELETE,  OR [I]
     NSERT"
5180 GOSUB 5540:IF A$="C" THEN 520
     0
5190 GOTO 5460
5200 GOSUB 9200
5240 IF W(V)=0 THEN 5270
5250 Y=ASC(NTOCT$(TP+1,TP+1)):O=IN
     T(Y/13):N=Y-INT(Y/13)*13
5260 PRINT "[K]NOTE NUMBER: ";V
5270 A$="":PRINT "NEW LENGTH:",:IN
     PUT A$:IF A$="" THEN 5140
5280 IF A$<>"/" THEN D=VAL(A$)
5290 A$="":PRINT "NEW NOTE:",:INPU
     T A$:IF A$="" THEN 5390
5300 IF A$<>"/" THEN N$=A$:IF IF LEN(
     N$)>2 THEN 5290
5302 IF LEN(N$)=1 THEN N$(2)=" "
5310 FOR N=0 TO 12:IF NT$(N*2+1,N*
     2+2)<>N$ THEN NEXT N:PRINT "?N
     OTE NOT FOUND.":GOTO 5290
5320 A$="":PRINT "NEW OCTAVE:",:IN
     PUT A$:IF A$="" THEN 5390
5325 IF VAL(A$)>2 THEN A$="2"
5330 IF A$<>"/" THEN O=VAL(A$)
5390 NT=N+O*12:FOR TP=0 TO 62:IF A
     SC(NTOCT$(TP+1,TP+1))<>NT THEN
      NEXT TP
5392 GOSUB 9500:REM PACK IT INTO W
     (V)
5400 V=V+1:GOTO 5200
5415 IF V<0 THEN V=0
5420 PRINT "[K]":GOTO 5030
5430 IF A$="P" THEN V=PP-60:GOTO 5
     415
5440 IF A$<>"N" THEN 5450
5442 POKE 703,24:PRINT "[K]":IF PP+2
     9>MAX THEN PP=MAX-30
5443 IF PP<0 THEN PP=0
5444 V=PP:GOTO 5030
5450 IF A$="C" THEN V=PP-30:GOTO 5
     415
5458 RETURN
5460 IF A$="D" THEN 5510
5480 FOR I=MAX TO V STEP -1:W(I+1)
     =W(I):NEXT I:MAX=MAX+1:GOTO 51
     40
5510 FOR I=V TO MAX:W(I)=W(I+1):NE
     XT I:MAX=MAX-1:GOTO 5140
5540 GET #1,A:A$=CHR$(A):IF A$=""
     THEN 5540
5550 RETURN
5999 END
6000 ? :? :? "          TEMPO = ";S
6010 ? "       NEW TEMPO = ";:IN
     PUT S
6020 IF S<1 THEN 6010
```

so that you can look for the note (or notes) that you wish to change. The 'Current' command simply redisplays the table starting at the same note. This option is usually used after an insert, delete, or change to see how the song looks.

The '#' option allows you to change the current notes in memory, one at a time. Note that the editor does not want the number sign itself; it is expecting a number. Also, it does not change any notes on the tape or disk, so don't worry about making a mistake. Experiment with changing notes and then play them together to see how they sound.

Pressing the return at the 'NEW LENGTH:' prompt will return you to the main menu. Pressing return at any other prompt will leave the rest of the parts of the note unchanged and skip ahead to the next note. This allows you to change the lengths without re-entering the remaining values that you don't want to change.

Pressing the '/' and return at any prompt preserves that one value and jumps to the next prompt. For example, if you don't want to change a value, such as length, but do want to change the note or octave, press '/' and it will skip over the values you want to save without your having to retype them.

The insert option moves all the notes (including the one whose number you entered) forward one position and then returns you to the 'Next, Previous, Current, or #' prompt. You might want to use the 'Current' option now to see how the song looks. Change the note that was at the point of insertion to whatever specifications you want and you have just entered a note.

Delete moves all the notes following the one you specified down one position, effectively erasing that note. As with the insert option, you will be returned to the 'Next, Previous, Current, or #' prompt. Again, you may want to use the 'Current' option to see how the song looks after the deletion.

When you are prompted for a new note, the only notes that will be accepted are 'C', 'C#', etc., as listed in the data statement (line 13010). A rest is indicated by '[ ]'. The program looks up your note in the note table and converts it into the number that represents that particular note. If it can't find that note, it will give you the '?NOTE NOT FOUND.' error and return to the 'NEW NOTE' prompt to give you another chance to change it.

Other Menu Options

There are five additional menu options that allow you to hear your song, load a song from or save a song to the tape player, change the tempo of the song, or quit.

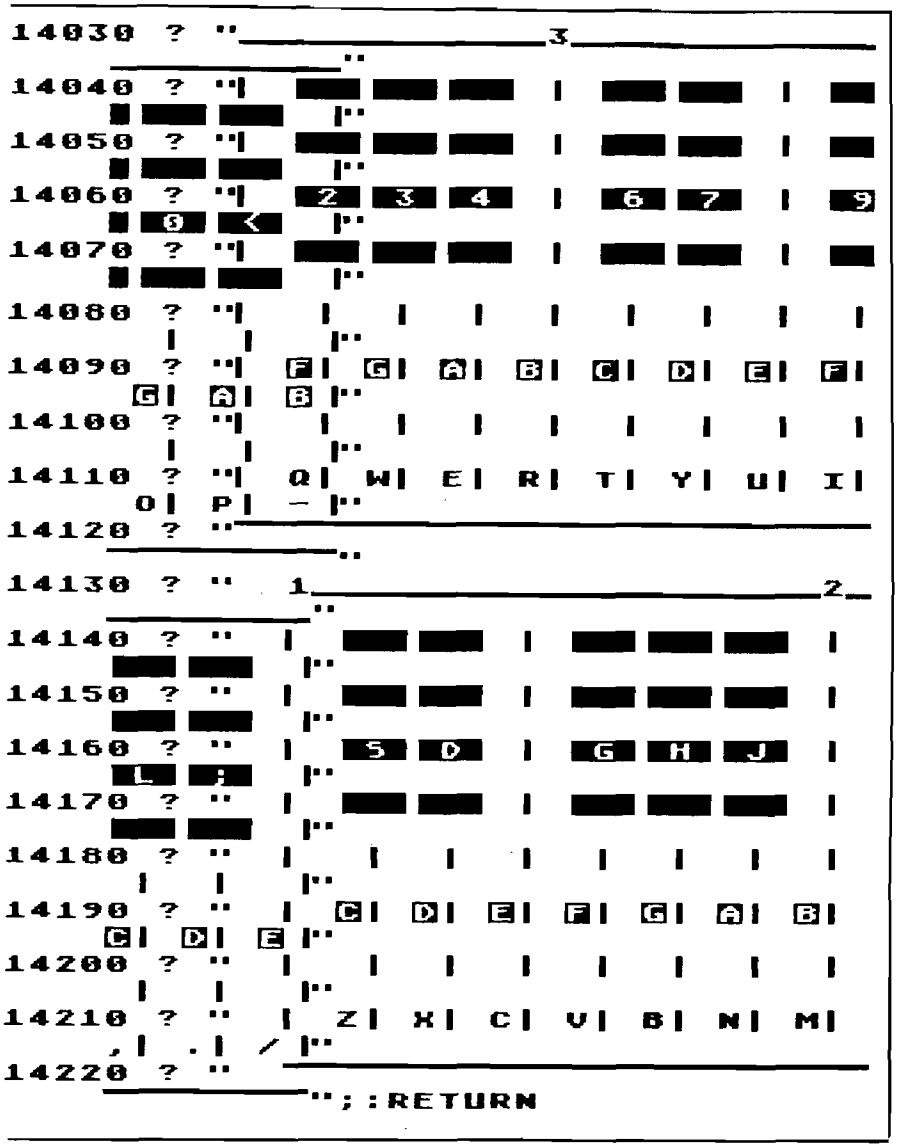
Choosing option 2, 'REPLAY SONG,' will play the song currently in memory over the television speaker. The routine uses the current tempo for the speed at which to play the song. If there are no notes in the song, then the error '?NO SONG IN MEMORY' is printed and you are returned to the menu.

'SAVE' and 'LOAD' options are numbered 3 and 4, respectively. To save the current song, choose option 3 and enter the file name of the song to be stored. If you are using a cassette you can specify only 'C:' as your file name. If you are using a disk, the file name must be a standard disk file name. Note that this file name and the name of the song you specify when you choose option 1 (PLAY SONG) are not the same. The name refers to the name that you wish the song to be titled — for the sake of convenience only. The file name is merely where you want the song to be stored and applies only to disk users. If you are using a cassette-based system you will be prompted to press PLAY and RECORD on the tape player. When the song is saved, the number of notes and the file in which they are saved are displayed and you are returned to the menu. If you are using the tape player, it will stop.

Option 4 is similar in operation to option 3. Remember that when you LOAD a new song, you will erase any song currently in memory; you must SAVE the current song (if you want to keep it for later use) before loading a new one. The file name prompt will now appear. (If you are using a cassette, there are no file names that you can specify. The tape will read from where it was left last.) When the song is loaded, as when saving, the number of notes and the file they were in are displayed and you are returned to the menu. Again, if applicable, the tape player will stop.

To change the current tempo setting, choose option 5. The minimum (fastest) setting allowed is '1'. There is no restriction on the

```
6030 RETURN
7000 CLOSE #1:GRAPHICS 0:END
8000 SOUND 0,0,0,0:GOSUB 9200
8010 ? "K":POSITION 3,3
8020 ? "CHOICES:":? :?
8030 ? "CONTINUE"
8040 ? "REPLAY"
8050 ? "BACK UP 1 NOTE":?
8060 ? "WHICH? ";
8070 GOSUB 5540:REM GET ROUTINE
8080 IF A$="C" THEN GOSUB 9200:SOU
     ND 0,0,0,0:RETURN
8090 IF A$="R" THEN GOSUB 2000:GOT
     O 8010
8100 IF A$<>"B" THEN 8070
8110 IF V=0 THEN 8070
8120 GOSUB 9200:GOSUB 9800:V=V-1
8130 SOUND 0,0,0,0:GOTO 8070
9000 FOR I=1 TO 300:NEXT I:RETURN
9100 PRINT :PRINT "C: for cassette
     , D:filename for    disk"
9105 PRINT "Enter filename...";:IN
     PUT F$
9110 V=1:IF F$<>"C:" THEN RETURN
9150 ? "INSERT SONG TAPE AMD PRESS
      RETURN"
9160 GET #1,A:? :POKE CF,0:RETURN
9200 TP=INT(W(V)/1000)
9210 D=W(V)-TP*1000:RETURN
9500 IF D>999 THEN D=999
9510 W(V)=INT(TP*1000+D):RETURN
9800 SOUND 0,ASC(PITCH$(TP+1,TP+1)
     ),D5,L
9810 FOR I=1 TO 5*D/8:NEXT I:RETUR
     N
10000 GRAPHICS 0:SOUND 0,0,0,0:POK
      E CF,1:SETCOLOR 1,0,0:SETCOLOR
       2,3,4:SETCOLOR 4,7,6
10030 ? "K↓↓↓                ATARI PLAY
     ER":? :? :V=0
10040 ? "              1 PLAY SONG"
10050 ? "              2 REPLAY SONG"
10060 ? "              3 SAVE SONG"
10070 ? "              4 LOAD SONG"
10075 ? "              5 EDIT SONG"
10080 ? "              6 CHANGE TEMPO"
10090 ? "              7 QUIT↓"
10100 POKE CF,0:? "              CHOOSE
     ? ";:RETURN
12000 S=10:LN=500:D5=10:L=8:CF=752
12010 DIM W(LN),PITCH$(63),NT$(26)
     ,F$(14),B$(20),N$(3),A$(3),NTO
     CT$(63)
12020 RESTORE :B$="NONAME"
12042 READ PITCH$,NT$,NTOCT$
12050 RETURN
13000 DATA r◤f♥♥♥♥♥(♥#/♥—→♥6♥A♥♥WY
     SD♥L9♥♥U♥y♥1&♥C♦♥@♥H5♥<Q[*♥%2♥
     ♥!♥♥ L♥♥-f
13010 DATA C C#D D#E F F#G G#A A#B
      [
13020 DATA _◣♣" ♥" ♥♥!♥#←♥→%♥┐♥┤♥♥◥ |
     ♥┬♥▂ᴱ♥♥┼♥‾♥▪ ▪♥ "r♥┴♥┤ ↑♥└▪— ♥"↓♥
     ♥$♥" ◢┘♥♥/├
14000 POKE 82,2:POKE CF,1:? "K↓PLA
     Y WHEN READY   SONG:";B$:SETCO
     LOR 1,0,0:SETCOLOR 2,12,6
14010 ? "SPACE BAR TO PAUSE, RETUR
     N TO END SONG";
14020 ? "A, F, OR K = REST"
```



maximum (slowest) setting, except for the fairly large internal limits of the Atari. However, even a setting of just 100 will result in extremely long notes.

When you are finished with Atari Player and want to return to BASIC, choose option 6. Always remember to save any song that you are currently working on before choosing this option. If you forget this rule, typing 'GOTO20' **might** enable you to return to the program without losing your song.

Programming Concepts

Using a Menu to Make a Choice

1. Input with a Single Keystroke

Normally input from devices is done with the INPUT statement. The only limitation of this statement is that it requires the input data to be followed by a carriage return. Thus, any input from the keyboard using this method would require at least two keystokes — one for the character being input and one for the return key. The way around this is illustrated in program lines 30 and 50.

Line 30 opens the keyboard for input. This means exactly what it seems: you can now input from the keyboard. The open statement is usually used for access to files on cassette or disk, but it can be used for any connected device, such as the keyboard and even the screen. The device specification for the keyboard is 'K:', which is the open statement's fourth parameter.

Because a single keystroke input is preferred, the keyboard input for the menu (line 50) is done with a GET. The GET statement waits for one character of data from the keyboard; i.e., the single keystroke. Thus no carriage return is required and an option may be selected by striking a single key. However, the data input is not a character, or a letter, but rather a number. This number is the numerical representation of the typed letter and is called an ASCII code.

2. ASCII Codes

Because computers cannot handle anything other than numbers, a method for handling characters such as punctuation marks, letters of the alphabet, or any other character, is needed. What has been devised is the ASCII codes (American Standard Code for Information Interchange), which are numerical representations of each character for the computer to handle internally. Most users never see these numbers for they are converted back to characters when displayed on the screen, but they are neccessary when dealing with such specific statements as the Atari GET.

3. Using a Menu to Make a Choice: Selection by Number

When you run Atari Player, the first display that you see is a list (or menu) that tells you what options are available. Each item on the menu is selected by pressing the number associated with it. The BASIC program steps required to evaluate your choice are in lines 50 and 60.

```
50  GET #1,A:
    IF A<49 OR
    A>55 THEN 50
60  A=A-48:GOSUB
    A*1000:GOTO 40
```

Line 50 gets the ASCII code for the keyboard character pressed and stores it in variable A. The following IF...THEN statement checks to see whether or not the character typed was a number. Note that, as stated above, the GET statement gets the internal code for the character typed rather than the character itself. This means that the program has to check for the characters 1-9 rather than the numbers 1-9. If A is less than 49 (ASCII 1) or greater than 55 (ASCII 7), then the character typed is ignored and the program returns to line 30 to get another choice from the keyboard.

Line 60 does the actual branching to the appropriate routine. Since the only options available are listed 1 through 7, and the program routines for each start on the line numbers that are multiples of a thousand (1000-7000), the branching to these routines can be done by formulae. All that is needed to branch to the appropriate routine is to multiply the option number chosen by 1000 and then perform a GOSUB to that result. Thus, 48 (ASCII 0) is subtracted from the code in A to convert it to one of the numbers 1-7. Then a GOSUB to the line number A * 1000 is performed.

This is a useful space conservation technique. The only other way to do this would be to have a fairly large ON...GOSUB statement followed by the starting line number of every subroutine.

4. Selecting by First Letter

If the space bar is pressed during the 'PLAY SONG,' then another menu is displayed.

The first character of each item is displayed in reverse video to indicate which letter is to be pressed on the keyboard to select that choice. The routine that gets the characters from the keyboard is at line 5540. Note that this routine converts the number put in variable A into a string (A$) *via* the CHR$ command. This is done so that the routine will return a string holding the choice as opposed to returning the ASCII value of the choice. That is, this routine returns a letter, which is then serviced by its own IF...THEN-statement.

```
8070 GOSUB 5540:REM GET ROUTINE
8080 IF A$=C THEN GOSUB 9200:SOUND
     0,0,0,0:RETURN
8090 IF A$=R THEN GOSUB 2000:GOTO
     8010
8100 IF A$<>B THEN 8070
```

If there are only a few choices, as in this example, then there is not a lot of code. If there were many choices, then the amount of code to service the letters could be significant.

5. Accessing the Keyboard for Real-Time Situations

The above method of selecting by first letter using the GET statement is fine for most situations involving input with a single keystroke, but it is severely limited in real-time applications. This is because the GET statement actually waits for a key to be struck. For instance, suppose you were writing a game that needed to keep track of the elapsed time between keystrokes. This would be impossible with the GET statement because the loop required to keep track of the time could never be executed. Such was the problem in writing the PLAY SONG option, which has to keep track of such things.

The Atari uses memory location 764 as the place to keep the last key hit. That is, whenever a key is struck (no matter what program is running) location 764 is filled with a number that represents the last key pressed. But this number is **not** ASCII. This location uses the actual hardwired codes for the keyboard — a different set of codes entirely. For the beginner, using this location is rather difficult as the codes follow no obvious pattern. For instance, a space is a 33, the letter A is 63, the B is 21, and so on. Therefore, the user must know which keys to expect ahead of time. The following short routine can be used to discover the various key codes. After typing RUN, simply hit a few keys; their corresponding codes will come up on the screen.

```
10 PRINT PEEK(764):GOTO 10
```

Lines 1030, 1060, and 1070 show how Atari Player uses this location. Note that it is POKEd with 255 after being read. This insures that the same key can be hit twice and registered as two keys. That is, because the location holds the last key hit, if you were to hit, say, the 'Z' key twice, and if the location was not reset after each read, it simply would not change from the first 'Z' to the next. This means that two notes would be read as one (with a rather long duration). Resetting location 764 also allows us to keep track of the duration in the first place (line 1030). By checking for the 255, we know that the key has not yet been hit and that the duration should then be incremented.

Space Saving with Arrays

Atari Player uses many arrays — for the storage of its notes, keyboard entries, and other such internal data. There are many ways of storing this information in the arrays. This section deals with the most economical way possible for storing the types of information contained in Atari Player.

1. Numeric Arrays

Atari Player could have stored all of its information in the form of numerical arrays as all of its information is, indeed, numbers. But numbers are very costly to store in BASIC arrays, especially on the Atari. Atari BASIC 'eats' up 15 bytes plus 6 bytes for every cell in the array. This is not including the space that is eaten up by the variable name and the DIM statement itself. For example, an array that was dimensioned for 100 cells would eat up 15 plus 6 times 100 — or 615 bytes! Therefore, the only array that we made numerical was W, and it was dimensioned for a whopping 501 cells. Over 3000 bytes is used to store it.

However, things could be worse. The W array has a little more information than normal crammed into it. In fact, that one array is made up of both the keystroke (for the note played) and the length of time that it is played. It was necessary to perform such a numerical combination to keep from wasting an exorbitant amount of memory. But before we get into such a space-saving trick, let's first examine how **not** to program the storage area.

As mentioned briefly before, each note that you play involves two pieces of information that must be saved by Atari Player: the number (keystroke) of the note and the duration of the note.

We could define a two-dimensional array that contains a number for both parts of each note: e.g., DIM W(500,1) would reserve space for 500 notes (the zero*th* element is not used) with two numbers per note. How much memory do you think this would take? It really is deceiving. Since 500 times 2 is 1000, is that the total number of bytes required? No! Each Atari BASIC array number requires six bytes of memory. Therefore, it would take 6 * 1000, or 6000, bytes of memory! On a 16K Atari (which, by the way, has only 13000 some bytes left after the operating system has taken its share) that leaves a little over 7K for the rest of the program.

You have to get a bit tricky to squeeze more out of the song space; but there is nothing wrong with getting tricky when writing programs. In fact, that can be half the fun! To really squeeze the memory in Atari Player, we took advantage of the size of the number that the Atari arrays can hold. The two values that we need to keep for each note played are the note number (0 to 38) and the duration (which is limited to 0 to 999 units). If only we could pack both of these individual values into a single number for storage

```
5392 GOSUB 9500:REM PACK IT INTO W(V)
          .
          .
          .
9500 IF D>999 THEN D=999
9510 W(V)=INT(TP*1000+D):RETURN
```

and then unpack them when we needed to use them. Well, good news — we can!

The technique to pack the numbers is shown in line 5392.
This equation is not as difficult as it may at first appear.

W (V) is the array of notes where V is the number of the note in the song

+TP * 1000 multiplies the keystroke number (0 to 63) by 1000

+D adds the duration value (0 to 999)

All we have done is multiply the keystroke value that we need to save by enough to make sure it does not overlap with the duration value. This insures that we will be able to unpack the separate parts later. The unpacking is a bit more difficult than the packing, but conceptually it is simple. All we need to do is reverse the packing process. This is accomplished in the following lines:

```
9200 TP=INT(W(V)/1000)
9210 D=W(V)-TP*1000:RETURN
```

Line 9200 restores the note number by dividing by 1000.

Line 9210 restores the duration by subtracting the note number multiplied by 1000. This simply returns the remainder that was sheared off by the INT 1000.

The above lines of program take the single integer value and convert it back into two separate parts. This method results in a 500-note song requiring only 501 array slots to store it, at six bytes per number, for a total storage of just over 3000 bytes — only half the amount that would be needed if we stored the two values separately!

2. Simulating String Arrays

NT$ is a string that holds all the possible note symbols. Normally these symbols would be entered separately in a string array, but the Atari does not have this capability. (It does dimension strings, but this refers only to the lengths of the strings and that doesn't help us much.) Instead, all string information must be concatenated, or 'strung together,' one after another in a single string. This means that the only way to access the different elements is to access the different character locations within the string. As long as the elements are of equal size this is fairly simple because the starting character position of any of the elements can be calculated using the index. Take for example the following lines:

```
10 DIM NAME$(30)
20 NAME$="TOM  DICK HARRYJIMBO
   RALPHBILLY"
100 INPUT INDEX: REM ---INDEX
   STARTS AT 0---
110 PRINT NAME$(INDEX*5+1,INDEX
   *5+5)
120 GOTO 100
```

This routine simulates a string array where the elements are first names. In Atari BASIC, to get part of a string, you specify the string name followed by the starting and ending character position of the substring you want. In line 110, what is printed is the substring that starts with character position INDEX*5+1 and ends with INDEX*5+5. Thus, because the character position is calculated, every element must be the same length. If they aren't, and this is usually the case, then the smaller ones must be padded with spaces. Every element must start at every character position that is a multiple of some number in order for this particular technique to work.

The note symbols used by Atari Player are stored the same way. They are stored initially in line 13010 and then READ in as a single string (NT$) in the initialization routine at lines 12000 through 12050. Assuming Y is the note number (0-12), the following routine will set N$ equal to the corresponding note symbol:

```
10 N$=NT$(Y*2+1,Y*2+2)
```

In fact, this very routine is found at the end of line 5062. Note that the elements in this case must be two characters long: a note, 'C' for instance, followed by either a space or a sharp sign, '#'.

There is another way that strings are used in Atari Player. It is a bit more complex but basically entails the same concepts. The difference lies in the fact that this method is used to store a series of numeric values such as would normally be done with numeric arrays. The savings are space and time and for the following reasons.

Strings use up only one byte of memory per character. A little more is used up in the variable name and such, but that is essentially it. So, if we were to somehow convert every number that is in a numeric array (six bytes) to a string character (one byte), we would have a savings of five bytes per element. Thus, a numeric array dimensioned for 100 elements that would normally eat up 600 bytes of storage could be made to fit in merely 100 bytes.

Loading in the initial values in a string is very fast, whereas loading in the initial values in an array is tediously slow. This is because string loading can be done with a single statement and all the elements can be loaded in at once; the elements in an array must be loaded in one after another in a loop. For arrays of ten elements or less, conversion to a string is hardly worth the effort because the loading is so short and, therefore, does not take up much time anyway. But for larger arrays, string conversion is definitely worth-

```
12000 S=10:LN=500:DS=10:L=8:CF=752
12010 DIM W(LN),PITCH$(63),NT$(26)
     ,F$(14),B$(20),N$(3),A$(3),NTO
     CT$(63)
12020 RESTORE :B$="NONAME"
12042 READ PITCH$,NT$,NTOCT$
12050 RETURN
13000 DATA r◤f♥♥♥♥♥(♥#/♥—→♥6♥A♥♥LY
     5D♥L9♥♥U♥y♥1E♥C◆♥@♥H5♥<Q[*♥%2♥
     ♥!♥♥ ■♥♥=f
13010 DATA C C#D D#E F F#G G#A A#B
     []
13020 DATA ▁◣²" ♥" ♥♥!♥#←♥→%♥┐♥┤♥♥\ |
     ♥┬♥▁E♥♥+♥¯♥. .♥ "┌♥┴♥┤ ↑♥└e— ♥"↓♥
     ♥$♥" ◢ ♥♥/├
```

while because the loading time of even several minutes can be cut down to only a few seconds at most.

Lines 12042, 13000, and 13020 demonstrate how the strings are stored and read in.

There are a couple of problems when working with these strings. First, when addressing the element you must, in reality, address the individual characters of the string. That is, the index that you would normally use to access any of the elements in the numeric arrays is now the character position in the string. But element positions in arrays start at zero (0) while string positions start at one (1). Therefore, when accessing the data in strings, you must either ignore the zero*th* element entirely and start the element numbers off at one, or add one to the element number so that the element that was previously numbered zero will reside at string position one (1). Line 9800 demonstrates how this latter method is used in Atari Player.

```
9800 SOUND 0,ASC(PITCH$(TP+1,TP+1))
     D5,L
```

where TP is the index to the string array PITCH$.

The second problem is a little tougher for the beginner to understand. Since each character in a string is essentially one byte of storage internally, the size of the numbers you can store are limited to what a byte can store. Because, as you may know, bytes are only eight bits long, the largest number that could possibly be stored in one is 11111111 binary (255 decimal). Also, the number must be a non-negative integer. To get the number in the string as an element it must be converted to a character *via* the CHR$ function. This function will turn the number specified into the character whose ASCII value is that number so that it may be stored in a string. When accessing that element, to get the number from the character, the ASC function must be used. This function converts the string specified (presumably an element in the string) to its ASCII equivalent. (See Programming Concepts 2: ASCII Codes.)

MICRO™

Atari Player Variable Usage

Constants

CF	POKE location for cursor inhibit
DS	Sound distortion
L	Sound loudness
LN	Last element available in W array
NT$	String 'array' of all note symbols
NTOCT$	Note/octave conversion from keystroke number
PITCH$	String 'array' of sound pitches for conversion from keystroke

Variables

A	ASCII value from keyboard
D	Duration for current note
I	Loop counter
MAX	Position of the zero element in the song (end of song)
N	Note number (0-12)
NT	Actual note (0-12) plus octave (0-2)
O	Note's octave
S	Tempo setting (speed of song)
TP	Keystroke/note equivalent
V	Note pointer for song
W( )	Array of notes and durations
XX	X position of editor's output
YY	Y position of editor's output
A$	Input string from keyboard
B$	Song name
F$	File name
N$	Note symbol

Sure it's insured?

SAFEWARE℠ Insurance provides full replacement of hardware, media and purchased software after a low $50 deductible. As little as $35/yr covers:

• Fire • Theft • Power Surges
• Earthquake • Water Damage • Auto Accident

Select the coverage you want from the table.

Amount of Insurance	Annual Premium
Up to $ 2.000	*$ 35*
$ 2.001-$ 5.000	*$ 60*
$ 5.001-$ 8.000	*$ 75*
$ 8.001-$11.000	*$ 90*
$11.001-$14.000	*$105*

Call for higher coverages.
Not avail. in AK, DC, HI, KY, LA, ME, MS, NV, SC, or WY.
Call for immediate protection.

1-800-848-3469

(In Ohio call 1-800-848-2112)

SAFEWARE™

COLUMBIA NATIONAL GENERAL AGENCY

MICRO™

From Here to Atari

by Paul S. Swanson

The number of new peripherals you can add to your Atari computer has been steadily increasing. The ATR8000 is a device produced by SWP Microcomputer Product, Inc. that adds CP/M and a few other interesting capabilities. Another addition is a modem, which opens up a new direction to microcomputer users. In addition to the modems which can be attached to the Atari computers directly, any RS232C modem can be used either with the 850 interface module or under CP/M with the ATR8000.

The ATR8000

This peripheral is actually a computer on its own and can be used with or without an Atari computer attached. It does have the standard Atari serial bus connectors, but it can use any RS232C terminal instead of an Atari, connected through its modem port.

I have had an ATR8000 now for about a month and use it to connect three Radio Shack disks to my Atari computer. With such a setup, the drives can be used as Atari drives when not in CP/M mode, or as dual density CP/M drives in CP/M mode.

Using the ATR8000 as a peripheral to the Atari computer (i.e., not in CP/M mode) allows mixing the drives so that drive 1 may be connected to the disk port of the ATR8000 and may be a Radio Shack or other non-Atari drive and drive 2 may be an Atari 810 or 1050 connected along the serial I/O bus. In the CP/M mode, an Atari 810 or 1050 disk drive may not be used.

Two other devices that can be directly connected to the ATR8000 are parallel printers and RS232 devices, such as a serial printer or a modem. When the ATR8000 is not being used for CP/M, it will act as a printer buffer for the printer connected to it. A 64K ATR8000 provides a 48K printer buffer. This buffer will work for any printer attached to the parallel port on the ATR8000. It does not work for printers that are connected using the Atari serial I/O bus or 850 interface module.

A modem can also be connected to the ATR8000 using its serial port. The modem must be RS232 compatible and is controlled directly by the ATR8000. The modem connected in this fashion is not available to the Atari computer as device "R:", but may be used under CP/M. A modem connected to the 850 interface may be used by the Atari computer as device "R:" when not in CP/M mode. There are several different configurations allowed with the ATR8000, particularly if you also have an 850 interface.

There are also a few drawbacks to consider. The device does generate a large amount of television interference, particularly if you use a ribbon cable connector for the printer. It may even display an interference pattern on the television you use as a monitor for your Atari computer. A ferric core is supplied with the ATR8000 to suppress this interference (by wrapping the cable from the Atari to the television around it) on your monitor/television. Another problem is that you have to build cables. The pinouts are provided in the manual and the three ports on the ATR8000 are edge card connectors. The printer cable required the most work to assemble. The disk cable was relatively easy. The Radio Shack disk cables will work with no problem, even when using disks which are not Radio Shack disks. The only caution with using Radio Shack disk cables is that the drive selection is done in the cable, so you have to put the drives onto the cable in the correct order and the select on the disk drive must agree with the select on the cable connector.

The ATR8000 manual explains how to alter a Radio Shack drive to use it on the ATR8000. I ran into a slight problem with that, however, because the Radio Shack drives I have are not the same kind as the ones described. If you try this and find you have the older Radio Shack drives, all that is required is to sever the only jumper across the connector inside the drive. When you open the drive, which wire to cut will be obvious.

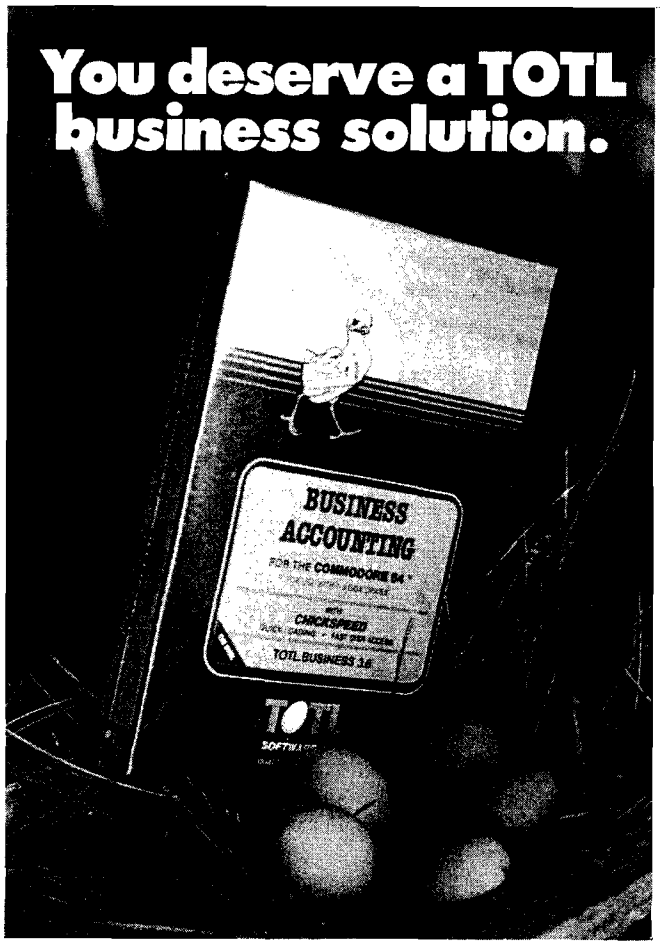
Since most "bare" disk drives will work with the ATR8000, including 8" drives with an alteration, it is tempting to get an ATR8000 and two drives from a surplus electronics store instead of two Atari drives if you are upgrading to a disk system. Careful shopping can result in drives for less than $150 each. The ATR8000 lists at $349.95 (16K without CP/M) or $499.95 (64K with CP/M), which adds up to $649.95 or $799.95 - less than the cost of the two Atari disk drives. You will still need the cable and a power supply for the disk drives, but if you do this, you will also have the printer connection with the 48K buffer and a modem port, plus CP/M if you get the 64K version, not to mention the ability to add a third drive for under $150 if the power supply is adequate for three disks.

Using CP/M with an Atari does have one serious problem associated with it, which is that CP/M requires an 80-column screen and the Atari screen has only 40 columns. The software that accompanies the ATR8000 contains a program that turns the Atari into a terminal for CP/M use. The screen is sectioned and moves back and forth to allow you to see the 80 columns with a 40-column window.

There is a better solution for more serious CP/M users. Austin Franklin has provided the software on a cartridge for the 800 to allow use of his 80-column board with CP/M. The system runs on the Atari 800 computer. The 80-column board and the software for using it for the ATR8000 is available from Austin Franklin Associates, Inc., 43 Grove Street, Ayer, MA 01432, (617) 772-0352. The 80-column board lists at $289.95 and the software allowing its use with the ATR8000 lists at $29.95.

Telecommunicating with your Atari Computer

Modems create one of the more unusual ways to meet people. All over the country there are free bulletin boards for computers to call. If you have a modem on your Atari



WORD PROCESSING
TOTL.TEXT

MAILING LIST AND LABELS
TOTL.LABEL

TIME MANAGEMENT
TOTL TIME MANAGER

KEYWORD CROSS REFERENCE
RESEARCH ASSISTANT

For Commodore 64™ and VIC 20™

Announcing the newest members of the family...

BUSINESS ACCOUNTING FOR **$95** (SUG. RETAIL)
TOTL.BUSINESS

SPELLING CHECKER FOR **$35** (SUG. RETAIL)
TOTL.SPELLER (64 only)

DATABASE MANAGEMENT FOR **$50** (SUG. RETAIL)
TOTL.INFOBASE

TOTL SOFTWARE, INC.
quality you can afford

Ask your dealer about TOTL Software or send in the coupon for further details and ordering information.

1555 Third Avenue, Walnut Creek, CA 94596
PLEASE SEND ME MORE INFORMATION ON TOTL SOFTWARE

Name: _____

Address: _____

_____ Zip _____

computer, you were probably informed amply about major national networks where you, for a fee, can tap in and take advantage of various services. In addition, there are many telephone numbers you can use to connect your computer to services that are completely free.

These free bulletin board services, or BBS's, are run mainly by private individuals on many different types of microcomputers. They provide a public message base, which usually have some interesting conversations that you can jump into, as well as other features. Many have downloading sections where you can obtain copies of public domain programs for your Atari. Some have "feature articles" which are the electronic equivalent of magazine articles. Most have private mail capability to allow you to communicate with specific other users if you have something private to discuss.

There are also games available for you to play on many of these BBS's while you are connected. Bulletin files may be read, which tell you things about the computer you have connected with or about some recent alterations in the services provided. Almost all of them also have a help file you can read, which tells you how to operate the BBS.

I am now involved with two of these BBS's. One I mentioned in a previous column, which is the Atari World section of the Outpost (617-259-0181). Another I have set up on my Atari computer. This one, Nite Lite, is on line from 6pm to 6am (eastern time) at 617-576-2426. Both support message bases and have bulletins, help screens and games. Both also answer at either 300 or 1200 baud and have lists of other boards in this area. In the near future, I will post a list of several boards in other areas of the country, so you will be able to call the number for Nite Lite (please take careful note of the hours) and get a list containing numbers that may be closer to you.

In many ways, these BBS's resemble the CB radios, which allowed communication between people who have never met. Some of the informality and lack of self-consciousness that CB radios created in conversations can be seen on the BBS's. The main difference between CB radio and the message bases of these BBS's is that the messages have to be typed and can be reviewed. It seems to result in more sophisticated conversation, almost always interesting and well worth reading.

In addition to idle chatter, discussions concerning national or local politics, critiques concerning news items and other discourse, the message bases are also a very good source of help for you concerning, among other things, your computer. If you are having a problem and don't know how to solve it, you are looking for some hardware or just some advice on making a decision, all you have to do is post a message on one of the local BBS's. You will probably have several replies within a few days and if your problem is interesting, you may even see messages where other people have started discussing it on their own.

The future of these free BBS's looks very promising. As an example of what can happen, both Nite Lite and the Outpost are members of BINEXNET, which is Boston INformation EXchange NETwork. One of the planned functions of this network is an exchange of mail (the messages) among the member BBS's. If this is implemented successfully, this network will be looking to tie into other networks around the country to exchange mail. The result, if successful, will be the electronic equivalent of letter mail, except there will probably be no charge.

Atari Reviews

Product Name: **VoiceBox II**
Equip. req'd: Atari 800 computer with 40K of RAM
BASIC Cartridge Disk Drive
Price: $124.95
Manufacturer: The ALIEN Group
27 West 23rd St.
New York City, NY 10010
(212) 741-1770

Description: *The VoiceBox II* is a voice synthesizer that connects in daisy chain fashion to the computer just as the disk drives, etc. do. It is provided with three disks. These disks have the demo program plus programs and dictionaries which allow the use of speech or singing in your BASIC programs. There are also several games included which take advantage of the speech abilities provided.

Pluses: The synthesizer may be used to include speech and/or music to any BASIC program. This can make educational programs much more interesting to younger children. You may also become a song writer and producer for the Atari. A talking, singing head is provided complete with lip movement to match the sounds. There are two versions, one is just the outline of a head with eyes, mouth, etc. while the other is full featured including hair. The latter was programmed by Jerry White and is quite good.

Minuses: The use of the *VoiceBox II* is not easy if you want the sound to be realistic. The speech is reminiscent of someone from a Spanish speaking country who is learning English. In order to play the work back, it is necessary to have a *VoiceBox II*connected to the computer. Thus anyone who wants to use your speaking or singing program would also have to have the synthesizer.

Documentation: An eighteen page manual is supplied. This is sufficient to use the program but it could be more informative. New computer owners would have a difficult time at first.

Skill level required: Intermediate computer user.

Reviewer: Richard E. Devore

Product Name: **Mail List**
Equip. req'd: Atari 400/800/1200XL with 48K Atari
BASIC disk drive printer
Price: $39.95
Manufacturer: MMG Micro Software
P.O. Box 131
Marlboro, NJ 07746

Description: *Mail List* program. This program will handle up to 600 names per disk. It uses one complete sector per record and allows up to 125 characters per record. Each record may contain separate entries for "First Name",, "Last Name", "Address 1", "Address 2", "City", "State", "Zip Code", "Reference 1", and "Reference 2". Features a random access search which claims to find any file on the disk in less than 1 second. Supports use of up to 4 disk drives and multiple files on the same or different data disks.

Pluses: Menu driven and easy to use. Sorts on "Name 2", "City", "State", "Zip Code", "Reference 1" in ascending or descending order. Prints labels with or without the two reference fields.

Minuses: You MUST read the directions carefully and be sure to use OPTION 7-END to end the session or your records may be lost.

Documentation: A twelve page booklet gives adequate instructions for the programs use.

Skill level required: Any computer owner with the prerequisite equipment.

Reviewer: Richard E. Devore MICRO™

C64-FORTH
for the Commodore 64

FORTH SOFTWARE FOR THE COMMODORE 64

C64-FORTH (TM) for the Commodore 64 - $99.95
- Fig Forth-79 implementation with extensions
- Full feature screen editor and macro assembler
- Trace feature for easy debugging
- 320x200, 2 color bit mapped graphics
- 16 color sprite and character graphics
- Compatible with VIC peripherals including disks, data set, modem, printer and cartridges
- Extensive 144 page manual with examples and application screens
- "SAVETURNKEY" normally allows application program distribution without licensing or royalties

C64-XTEND (TM) FORTH Extension for C64-FORTH - $59.95
(Requires original C64-FORTH copy)
- Fully compatible floating point package including arithmetic, relational, logical and transcendental functions
- Floating point range of 1E+38 to 2E-39
- String extensions including LEFT$, RIGHT$, and MID$
- BCD functions for 10 digit numbers including multiply, divide, and percentage. BCD numbers may by used for DOLLAR.CENTS calculations without the round-off error inherent in BASIC real numbers.
- Special words are provided for inputting and outputting DOLLAR.CENTS values
- Detailed manual with examples and applications screens

(Commodore 64 is a trademark of Commodore)

TO ORDER - Specify disk or cassette version
- Check, money order, bank card, COD's add $1.50
- Add $4.00 postage and handling in USA and Canada
- Mass. orders add 5% sales tax
- Foreign orders add 20% shipping and handling
- Dealer inquiries welcome

PERFORMANCE MICRO PRODUCTS
770 Dedham Street. S-2
Canton, MA 02021
(617) 828-1209

CoCo To Smartmodem

By John Kelty

The Radio Shack Color Computer is certainly my favorite personnel computer, but the RS-232 connections are not exactly what I had in mind when I read that it had a serial interface (how many other RS-232 interfaces do you know of that use the data input line as the printer busy?). In any case, there are ways around these things, and to connect the Hayes Smart Modem to the CoCo, I have written this BASIC program (including a few POKES) to change the RS-232 input/status line response. With this change, the many automatic features of this fantastic modem can be programmed from Color Computer BASIC.

I am presently using a "64K" CoCo with Extended Disk Color BASIC, two disks, and the "Solution" (from Frank Hogg Labs). Since the program is fairly simple, I am sure it can be modified for whatever CoCo configuration you have. Most of the lines that are optional are listed as comments.

The interface cable is made with a DIN plug at the CoCo end and a male DB-25P connector at the Smartmodem end. The pin connections are described early in the program listing. You need 4-conductor cable and I have used as long as about 25 feet without any problems. It is convenient however, to have the Smartmodem near your computer setup, then you can hear the dialup and the send and receive carrier tones when you begin communicating. There are many programming options you can choose with the Hayes Smartmodem, and these are extremely well documented in the excellent Owner's Manual. (In case you have not guessed, I consider the Hayes Smartmodem my best computing purchase to date!).

The BASIC program is menu driven for ease of operation. If you BREAK and then RUN the program again, the Smartmodem functions are cleared ("AT Z" is sent to the modem). To write to the Hayes simply make the CoCo printer busy signal constantly high. I do this with software (remember that the modem is sending back information on that line as the data to CoCo) by programming the internal 6821 bit to be an output. Not to worry! Remember somebody said that you cannot type in anything into your Color Computer that will damage it (See some past Rainbow editorials). This programming is accomplished with the following:

```
A65314: BA1
POKE B,51: POKE A,1: POKE B,55:
POKE A,4
```

Now you can talk to your modem just like it was a printer with PRINT-2, "your modem command".

To get back into the CoCo original RS-232 mode, use the following:

```
(using same A and B as above)
POKE B,1: POKE A,0: POKE B,55:
POKE A,5
```

Now you can use a communications package such as the Radio Shack Videotex or a similar program (see MODCOMM elsewhere in the CoCo section) to communicate (the Videotex will set the data rate at 300 Baud as required by the Hayes 300 model).

I know little to nothing of amateur radio operations, but with these POKES, a few lines of BASIC, and possibly some hardware, the Hayes Smartmodem can be made to send morse code. Auxiliary relay contacts are provided in the modem to key the transmitter automatically. It cannot however, be made to directly receive morse code. Diagrams describing the modem to radio interface are found in the Smartmodem User Manual Appendix H. Obviously some nice commercial terminal packages exist for the CoCo, but this program has made me happy for some time now. With the Hayes Smartmodem and the Color Computer, I am sure you will also enjoy computer communications.

MICRO™

Listing 1

```
10 'SMART300
20 'KELTY ENGINEERING
30 'WRITTEN BY JOHN R. KELTY
40 'LINCOLN, NEBRASKA  68505
50 '
60 'USE THIS PROGRAM TO CONNECT
70 'AND PROGRAM THE HAYES
80 'SMARTMODEM TO THE RADIO
90 'SHACK COLOR COMPUTER.
100 '
110 'THE CONNECTOR CABLE SHOULD
120 'CONNECT TO THE RS-232 PORT
130 'IN THE FOLLOWING WAY:
140 '
150 'COLOR DIN   HAYES CONNECTER
160 '
170 '  1  CARRIER DETECT  8
180 '  2  RECEIVED DATA   3
190 '  3  SIGNAL GROUND   7
200 '  4 TRANSMITTED DATA 2
210 '
220 'THE 300 BAUD HAYES MODEM
230 'MAY BE USED WITH VIDEOTEX
240 'OR ANY OTHER TERMINAL
250 'PACKAGE THAT ALLOWS THE
260 'HAYES TO RUN AT 300 BAUD.
270 '
280 'THE HAYES WILL OPERATE
290 'LOCALLY AT HIGHER RATES
300 'BUT NO HIGHER THAN 300
310 'WHILE ON LINE.
320 '
330 'SEE YOUR USER'S MANUAL
340 'FROM HAYES FOR DETAILS.
350 '
360 '*****************************
370 '
380 'RESET PRINTER BUSY TO WRITE TO HAYES
390 '
400 A=65314:B=A+1:POKE150,180
410 POKEB,51:POKEA,1:POKEB,55:POKEA,4
420 DIM M$(5),SN$(9)
430 '*****************************
440 'FOR CASSETTE OPERATION, YOU
450 'CAN DELETE THE DISK CHECK,
460 'BUT IT COULD BE LEFT IN,
470 'WITH OR WITHOUT DISKS.
480 '*****************************
490 DK=0:'CASSETTE OPTION
500 IF PEEK(49152)=68 AND PEEK(49153)=75 THEN DK=1
505 'CHECK FOR RS DISK ROM
510 CLS:PRINT:PRINT"    ***** SMART MODEM *****"
520 PRINT"     BY: KELTY ENGINEERING"
530 PRINT"    ************************"
540 PRINT:PRINT"CONNECT CABLES TO MODEM,"
545 PRINT"COMPUTER AND PHONE COMPANY."
550 PRINT:PRINT:PRINT"(BREAK AND LIST THIS PROGRAM TO"
555 PRINT"SEE CABLE DETAILS)."
560 PRINT:PRINT:INPUT"PRESS ENTER WHEN READY";A$
570 CLS:PRINT
580 PRINT#-2,"AT Z":'CLEAR MODEM
590 '*****************************
600 'MENU
610 M$(1)="AUTO DIALING"
620 M$(2)="AUTO ANSWERING"
630 FOR S=1 TO 2
```

```
640 PRINTS;M$(S)
650 NEXT S
660 PRINT:PRINT:PRINT"SELECT ONE OF THE ABOVE"
670 A$=INKEY$:IF A$=""THEN670
680 ON VAL(A$) GOSUB 700,1310
690 GOTO 510
700 '*****************************
710 'AUTO DIALING ROUTINE
720 '*****************************
730 '
740 CLS:PRINT:PRINTM$(1)
750 PRINT:PRINT:PRINT"1 DIAL A SAVED NUMBER"
760 PRINT"2 DIAL A NEW NUMBER"
770 A$=INKEY$:IFA$=""THEN770
780 ONVAL(A$) GOTO800,980
790 GOTO740
800 '************************
810 HM=2:'HOW MANY NUMBERS
820 'TO CHANGE THIS LIST, MAKE HM=HOW MANY # YOU WANT TO
    LIST (MAX OF 9).
830 'THEN INSERT LINES SIMILAR TO THOSE BELOW WITH YOUR
    NUMBERS AND NAMES.
840 CLS:PRINT"NUMBERS YOU CAN EASILY CALL":PRINT
850 SN$(1)="1   XXX-XXXX          A FRIEND "
860 SN$(2)="2   XXX-XXX-XXXX      A BBS    "
870 FOR K=1 TO HM
880 PRINT SN$(K)
890 NEXT
900 PRINT:PRINT"SELECT ONE OF THE ABOVE"
910 PRINT"(BREAK AND LIST TO EDIT THESE   NUMBERS)."
920 A$=INKEY$:IF A$=""THEN920
930 NB=VAL(A$)
940 IF NB<1 OR NB>HM THEN 800
950 P$=MID$(SN$(NB),5,16)
960 GOSUB1090
970 RETURN
980 '*****************************
990 PRINT:PRINT"DIALING A NEW NUMBER"
1000 PRINT:PRINT"TYPE IN THE PHONE NUMBER THAT"
1005 PRINT"YOU WISH TO CALL."
1010 PRINT"EXAMPLES:"
1020 PRINT"1-800-XXX-XXXX    LONG DISTANCE
1030 PRINT"112-800-XXX-XXXX  FROM LINCOLN
1040 PRINT"XXX-XXXX          LOCAL CALL
1050 PRINT
1060 LINE INPUT"TYPE DESIRED NUMBER";P$
1070 GOSUB 1090
1080 RETURN
1090 '*****************************
1100 'NUMBER KNOWN AT THIS POINT
1110 'READY TO DIAL
1120 PRINT#-2,"AT F0 D"P$
1130 'ENABLE RS232 RECEIVE BIT
1140 POKE8,1:POKEA,0:POKE8,55:POKEA,5
1145 STOP
1150 IF DK=0 THEN 1230
1160 '*****************************
1170 'YOU CAN LOADM ANY TERMINAL
1180 'PACKAGE THAT YOU WISH BY
1190 'EDITING THE NEXT LINES.
1200 LOADM "MODCOMM"
1210 FOR D=1 TO 2000:NEXT D:'DISK MOTOR DELAY
1220 'EXEC
1230 CLS:PRINT:PRINT"YOU NEED TO USE YOUR OWN"
1240 PRINT"TERMINAL PACKAGE AT THIS POINT"
1250 PRINT"(SUCH AS VIDEOTEX AND OTHERS)."
1260 PRINT"IF YOU HAD PRELOADED A TERMINAL"
1270 PRINT"PACKAGE WRITTEN IN MACHINE"
1280 PRINT"LANGUAGE, THEN SIMPLY TYPE EXEC"
1290 PRINT"AND PRESS ENTER."
1300 GOTO 2040
1310 '**********************
1320 'AUTO ANSWER ROUTINE
1330 '**********************
1340 '
1350 CLS:PRINT:PRINTM$(2)
1360 PRINT
1370 'MENU
1380 AN$(1)="DO NOT ANSWER"
1390 AN$(2)="ANSWER IMMEDIATELY"
1400 AN$(3)="ANSWER AFTER XX RINGS"
1410 FOR SA=1 TO 3
1420 PRINTSA;AN$(SA)
1430 NEXT SA
1440 PRINT:PRINT:PRINT"SELECT ONE OF THE ABOVE"
1450 A$=INKEY$:IF A$=""THEN 1450
1460 CLS:PRINT:PRINT"        "M$(2)
1470 ON VAL(A$) GOSUB 1490,1560,1720
1480 GOTO 1310
1490 '**************************
1500 'DO NOT ANSWER
1510 PRINT:PRINT"      ###"AN$(1)"###"
1520 PRINT:PRINT"THE SMART MODEM ANSWER FUNCTION"
1530 PRINT"IS DISABLED."
1540 PRINT#-2,"AT S0=0"
1550 GOTO 2040
1560 '**************************
1570 'ANSWER IMMEDIATELY
1580 PRINT"   ###"AN$(2)"###"
1590 PRINT"THE SMART MODEM WILL ANSWER ALL"
1600 PRINT"CALLS IMMEDIATELY."
1610 PRINT"USE THIS TO TRANSFER CALLS THAT"
1620 PRINT"ARE ALREADY IN PROGRESS BETWEEN"
1630 PRINT"TWO INDIVIDUALS TO COMMUNICATION"
1640 PRINT"BETWEEN THEIR COMPUTERS."
1650 GOSUB 1870
1660 PRINT#-2,"AT"DP$"A"
1670 PRINT"AT THIS POINT YOU NEED TO RUN"
1680 PRINT"OR EXEC YOUR OWN RECEIVER"
1690 PRINT"PROGRAM TO INTERPRET THE"
1700 PRINT"INCOMING CALL."
1710 GOTO2040
1720 '**********************
1730 'ANSWER AFTER XX RINGS
1740 PRINT:PRINT"  ###"AN$(3)"###"
1750 PRINT:PRINT"THE SMART MODEM WILL ANSWER ALL"
1760 PRINT"CALLS AFTER THE NUMBER OF RINGS"
1770 PRINT"YOU SELECT (FROM 1 TO 255 RINGS)"
1780 PRINT
1790 INPUT"HOW MANY RINGS (1 TO 255)";RG
1800 RG=INT(RG)
1810 IF RG<1 OR RG>255 THEN 1790
1820 PRINT:PRINT"THE PHONE WILL BE ANSWERED"
1830 PRINT"AFTER"RG"RINGS."
1840 GOSUB 1870
1850 PRINT#-2,"AT"DP$"S0="RG
1860 PRINT:GOTO 1670
1870 '*****************************
1880 'FULL OR HALF DUPLEX
1890 PRINT:PRINT"DO YOU WANT FULL OR HALF DUPLEX?"
1900 PRINT"(IF YOU DO NOT KNOW, TRY HALF)."
1910 PRINT:PRINT"1    FULL DUPLEX"
1920 PRINT"2    HALF DUPLEX"
1930 A$=INKEY$:IF A$=""THEN1930
1940 ON VAL(A$) GOTO 1950,1990
1950 'FULL DUPLEX
1960 DP$="F1"
1970 PRINT:PRINT"FULL DUPLEX SELECTED."
1980 GOTO 2020
1990 'HALF DUPLEX
2000 DP$="F0"
2010 PRINT:PRINT"HALF DUPLEX SELECTED."
2020 PRINT
2030 RETURN
2040 '*****************************
2050 'EXIT ROUTINE
2060 'ENABLE RS232 RECEIVE BIT
2070 POKE8,1:POKEA,0:POKE8,55:POKEA,5
2080 PRINT:PRINT"THIS PROGRAM HAS ENDED."
2090 EXEC
2100 'TO USE THIS PROGRAM WITH MODCOMM
2110 'YOU MUST ADD A SWITCH TO THE MODEM
2120 'CABLE SO THAT PIN 1 CAN BE CONNECTED
2130 'TO PIN 2 AFTER DIALING THE NUMBER
2140 'SEE MODCOMM ARTICLE FOR EXPLAINATION
2150 END
```

MICRO™

MODCOMM

The Color Computer and Modem to Modem Communication

MODCOMM explains how to use the Color Computer to communicate to large mainframe computers. The program has the Color Computer emulating a dumb terminal.

by Walter Charlton

Modem to modem communication is often a necessary use of the Color Computer. Radio Shack has a ROM pack that will make the Color Computer emulate a dumb terminal; however, since the package is in ROM, there is no way to add other applications or features to the package. The purpose of this paper is to show how simple it is to write your own applications. Additionally, there is a program with this paper that will do most things the Radio Shack package will. The paper also shows the use and advantages of interrupts.

The program listed here is completely relocatable, so for those CoCo owners that only have 4K of memory, you can load the object code beginning at any location and it will work. The source code in the listing will take at least 16K of memory, if you use an assembler package.

The CoCo use the Peripheral Interface Adapter (PIA) to communicate to external devices. While the PIA is an excellent device, it was actually designed for parallel communication. When it is used for serial communication (for instance with the RS-232C), the information must be passed to the adapter one bit at a time. Timing loops must be designed to wait the proper bit time for baud rate timing.

There is also a drawback in the software. The processor must be totally dedicated to input or output at the time of communication, so a great deal of processing time is wasted. For instance, if the processor is bringing in information from the RS-232C I/O port, it cannot be interrupted for any reason or timing will be wrong and data will probably be lost.

The hardware configuration of the RS-232C I/O port is as follows:

Pin	Purpose
1	Carrier detect - CA1 Interrupt of PIA1
2	Data in - Bit 0 of Address &HFF22 on PIA1
3	Ground
4	Data out - Bit 1 of Address &HFF20 on PIA1

As the interrupt is currently wired (pin 1), it is useless for modem communication. The carrier detect just checks for carrier on the modem. If carrier exists, the interrupt is active (carrier is the medium that actually transmits the data over the phone lines). This would mean that communication must be totally software controlled. Better use of this interrupt would enhance the computers ability to communicate to a mainframe.

An interrupt can be used to force the MPU to a service routine. For instance, the computer can be busy doing anything and still be programmmed so that when an interrupt occurs, it will drop what it is doing and go to a vector address that will service the interrupt. After servicing the interrupt, it can then return to exactly what it was doing before the interrupt. Since data coming onto the computer will always be at random intervals, I wanted to use the interrupt line to bring in the data. I did this simply by tieing the data in line to the interrupt so that when a bit enters the computer, it will force an interrupt. This is a very simple technique.

I suggest that you first purchase a new serial I/O cable from Radio Shack. This cable should be used only for modem to modem communication. Select one end and lift a small tab out of its slot which will allow you to pull back the outer shield. The inner shield can now be separated which will bare the wires. Cut the wire to pin 1 (the pins are numbered on the front) and solder a small piece of wire from pin 1 to pin 2. After putting it all back together, mark the modified end. This end will be the one that plugs into the CoCo. We are now ready to start on the software.

Software

Without the use of the interrupt, the data in line would need continual checking. The interrupt significantly reduces the software needed for communication. I will first discuss the output routine and then the input. Most of the programming of the PIA is done on power up of the computer so nothing more is needed on that.

Lines 1700-1900 are the transmit routine. All that is needed to transmit is a store command to the address &HFF20.The data is held in register A of the MPU as shown in the DATAOT subroutine. Bit 0 is then shifted into the carry bit with the LSRA command. This is followed by two rotate left commands that position the bit into the transmit location of the B register. The B register is then stored into &HFF20 which transmits the data. After transmitting the entire byte, two stop bits are normally transmitted which are logic ones. Often one of the stop bits is not used. Also, some systems use parity, so if the program listed here does not work, check with your systems people to get what is needed.

You should note the branch to the subroutine WAIT in DATAOT. This is the baud rate timer. This routine can be used for both transmit and receive, and is currently set for 300 baud. For reasons I will explain later, it cannot be set for a baud rate higher than 300 in the program listed here.

The loop value for the baud rate timer is found by the following steps. First, find the cycle time of the CoCo by dividing the clock time (.895 MHz) into one. Second, find the baud rate time by dividing the baud rate (300 baud) into one. Third, divide the computer cycle time into the baud rate time to find the number of cycles for each bit time. Finally, the wait loop takes exactly 8 cycles, so after taking you the overhead (about 10 cycles), divide the above value by eight and the answer is the time for one bit wait. This procedure works for any baud rate.

Before discussing the receive technique, I will talk about how to set up the interrupt routine. Lines 640-720 show the routine that prepares the computer for interrupts. Bit 0 is forced high in address &HFF21. This is the control address for the A side of PIA1. By forcing the bit zero high, the interrupt is activated when CA1 is forced low by the peripheral. The FIRQ interrupt is tied to the PIA so the jump vector address is in &H110 and &H111. These addresses are loaded with the start address of the subroutine DATAIN which is the service routine for the interrupt. The final job of the initializing routine is to able the interrupt in the condition code register so that the interrupt will be recognized by the MPU. When an interrupt occurs, the MPU will go to addresses &H110 and &H111 to get the service routine address and will then execute it. After executing the service routine, and RTI command will return the processor to the point before the interrupt.

Subroutine DATAIN (lines 1230-1610) works just the reverse of subroutine DATAOT. The first thing that needs to be done when entering the subroutine is mask any interrupts. If this is not done, the computer will start interrupting interrupts as data comes into the computer. Another thing that needs to be done upon entering the routine is to save the state of all registers. The FIRQ interrupt will only save the program count and condition code register, so the rest of the registers must be saved to assure that nothing will be lost when returning.

The HBWAIT routine is a half bit wait. It is necessary to center the reception so that slight errors in the timing will not matter., Address &HFF22 is the data address for receive. The data is in bit 0 and after a load of the data register, two rotate commands are needed to position the data in the A register. The RORB moves the data into the carry bit and the second shifts the data into the A register. Since the interrupt is the CA1, it is necessary to read the PIA data register A (&HFF20) to clear then interrupt. This should be done just before returning.

A further comment needs to be made on interrupts if using a program pack for assembly. The program pack forces a continual interrupt that cannot be cleared. If this is not disabled, the program cannot be executed with the pack installed because the interrupt will always be active. The simplest way to disable the interrupt is by putting scotch tape over pins 7 and 8 of the program pack. These pins can be found by viewing the pack from the top. Hold it with the plug at the top and points 7 and 8 are the first ones on the top and bottom with copper runs from the right side. You can tell if you have the proper end because just next to pin 7 is the 5 volt pin which is slightly indented. After installing the program pack, it can be executed by typing EXEC 49152.

The Program

The program has a few features that have not yet been mentioned. It will output nearly the full spectrum of control characters. The downward arrow acts as the control key. It has a backspace key that when used with the shift key, will output the delete/rub character.Both lower and upper case characters are recognized, displayed and output (lower case letters have a dark background). By using the shift and clear keys, the program will be exited. Without the shift key, the clear key outputs the escape character. The break key will output the control C character. Shift zero will set the keyboard to lower case characters and output upper case only when the shift key is used.

This program does have a few weaknesses. The program cannot send and receive data at the same time. This would take a great deal of software and has little value so I did not put it in. I did, however, mask the interrupts going into the data out routine. This will allow the operator to transmit data even if a long receive message is coming in. One or two characters may be lost, however, on receive.

There will be some difficulty in increasing the baud rate on this program. Screen control and scrolling are done during the two stop bit times. Since screen control takes a great deal of time, any increase in the baud rate will result in garbage on there screen during scroll. As it was, I had to reduce the half bit wait to get the screen control in. If it is necessary to communicate at a higher baud rate, the received information will have to be buffered and displayed when the processor has more time.

The program, as listed here, is just one use of the Color Computer's ability to communicate to mainframe. Other uses could be for preloaded programs for later transmissions, downloading from a computer or use as an intelligent terminal. Happy computing! MICRO

Walter Charlton has his MBA from the University of Nevada, Las Vegas. He teaches computer programming part time and works as a Field Engineer for a major computer manufacturing firm.

Listing 1

```
                        00100 *
                        00110 *
                        00120 *      MODCOMM
                        00130 *
                        00140 *
                        00150 *
                        00160 *
                        00170 *  THIS PROGRAM EMULATES A DUMB TERMINAL.
                        00180 *  AFTER ASSEMBLY JUST EXECUTE AT THE BEGINNING
                        00190 *  OF THE PROGRAM.  TO EXIT PRESS SHIFT AND CLEAR.
                        00200 *  SERIAL I/O PORT MUST HAVE THE INTERRUPT TIED TO
                        00210 *  THE DATA IN LINE FOR INTERRUPT SERVICE.
                        00220 *  PROGRAM IS SET FOR ONE START AND TWO STOP BITS.
                        00230 *  THERE IS NO PARITY.
                        00240 *
                        00250 *
                        00260 *
                        00270 * WRITTEN  WALTER R. CHARLTON
                        00280 *
                        00290 *
                        00300 * MODULE INTERR ABLES THE INTERRUPS
                        00310 * AND INITIALIZES THE SCREEN
                        00320 *
                        00330 *
283B 1A   50            00340 INTERR  ORCC    #$50  DISABLE INTERRUPTS
283D 8E   6060          00350         LDX     #$6060  LOAD CLEAR CHAR
2840 108E 0400          00360         LDY     #$400  LOAD START ADDRESS
2844 AF   A1            00370 CLRS    STX     ,Y++  STORE CHARACTER
2846 108C 0600          00380         CMPY    #$600  DONE?
284A 23   F8            00390         BLS     CLRS  BRANCH IF NOT
284C 30   8D 01CF       00400         LEAX    TABLE,PCR  NAME PROGRAM
2850 108E 0400          00410 NAME    LDY     #$400  NAME START ADDRESS
2854 5F                 00420         CLRB            CLEAR B
2855 A6   80            00430 LDT     LDA     ,X+   GET A CHAR
2857 81   20            00440         CMPA    #$20  SPACE CHAR?
2859 26   02            00450         BNE     SP    NO? BRANCH AROUND
285B 86   60            00460         LDA     #$60  LOAD A BLANK
285D A7   A0            00470 SP      STA     ,Y+   STORE IT
285F 5C                 00480         INCB            INCREMENT COUNTER
2860 C1   08            00490         CMPB    #$8   DONE?
2862 26   F1            00500         BNE     LDT   NO? DO IT AGAIN
2864 30   8D 01BF       00510 LDT2    LEAX    TBLE,PCR  GET PROGRAMMER NAME
2868 108E 041F          00520         LDY     #$41F  GET START ADDRESS
286C 5F                 00530         CLRB            CLEAR COUNTER
286D A6   80            00540 CRDT    LDA     ,X+   GET A CHAR
286F 81   20            00550         CMPA    #$20  SPACE CHAR?
2871 26   02            00560         BNE     SP1   NO? GO AROUND
2873 86   60            00570         LDA     #$60  LOAD A BLANK CHAR
2875 A7   A0            00580 SP1     STA     ,Y+   STORE IT
2877 5C                 00590         INCB            INCREMENT COUNTER
2878 C1   14            00600         CMPB    #20   DONE?
287A 26   F1            00610         BNE     CRDT  NO? DO IT AGAIN
287C CC   05E0          00620         LDD     #$5E0  LOAD SCREEN START ADDRESS
287F FD   0088          00630         STD     >$88  STORE IT
2882 B6   FF20          00640         LDA     $FF20  CLEAR ANY INTERRUPTS
2885 8E   FF20          00650         LDX     #$FF20  LOAD ADDRESS OF PIA1
2888 A6   01            00660         LDA     1,X   GET CONTROL ADDRESS
288A 8A   01            00670         ORA     #1    FORCE BIT ZERO HIGH
288C A7   01            00680         STA     1,X   ABLE THE INTERRUPT
288E 30   8D 004B       00690         LEAX    DATAIN,PCR     LOAD SERVICE ROUT ADDR
2892 BF   0110          00700         STX     $110  STORE IT AT VECTOR
2895 B6   FF20          00710         LDA     $FF20  CLEAR INTERRUPT
2898 1C   BF            00720         ANDCC   #$BF  ABLE CONTROL REGISTER FIRQ BIT
                        00730 *
                        00740 *
                        00750 *
                        00760 *  THE FOLLOWING ROUTINE GETS A KEY DEPRESSION FROM
                        00770 *  POLCAT.  IT AFTERWARD DOES SOME KEYBOARD EDITING.
                        00780 *  ALL REGISTERS ARE MODIFIED AND THE DATA IS IN
                        00790 *  THE A REGISTER.
                        00800 *
                        00810 *
                        00820 *
289A AD   9F A000       00830 LOOP1   JSR     [$A000] CHECK FOR KEY
289E 27   FA            00840         BEQ     LOOP1   BRANCH IF NONE
28A0 81   13            00850         CMPA    #$13
28A2 26   04            00860         BNE     NOCMFL
28A4 86   60            00870         LDA     #$60
28A6 20   2B            00880         BRA     OUT
28A8 81   5C            00890 NOCMFL  CMPA    #$5C    CHECK FOR "\" CHAR
28AA 27   08            00900         BEQ     CMDLVL  YES? RETURN TO COMMAND LEVEL
28AC 81   15            00910         CMPA    #$15    CHECK FOR NAK CHAR
28AE 26   13            00920         BNE     NOSTOP  NO? GO AROUND
28B0 86   7F            00930         LDA     #$7F    MAKE A DEL CHAR
28B2 20   0F            00940         BRA     NOSTOP  GO AROUND STOP
28B4 8E   A0F6          00950 CMDLVL  LDX     #$A0F6  RESET INTERRUPT ADDRESS
28B7 BF   0110          00960         STX     $110  STORE IT
28BA B6   FF21          00970         LDA     $FF21  GET PIA CONTROL
28BD 84   FE            00980         ANDA    #$FE    FORCE BIT 0 LOW
28BF B7   FF21          00990         STA     $FF21  STORE TO DISABLE PIA INTERRUPT
28C2 3F                 01000         SWI             STOP
28C3 81   0A            01010 NOSTOP  CMPA    #10   CONTROL CHAR?
28C5 26   0C            01020         BNE     OUT     NO? GO TRANSMIT
28C7 AD   9F A000       01030 LOOP2   JSR     [$A000] GET ANOTHER CHAR
28CB 27   FA            01040         BEQ     LOOP2   KEY YET?
28CD 81   41            01050         CMPA    #65     CONTROL RANGE?
28CF 2D   02            01060         BLT     OUT     NO? GO TRANSMIT
28D1 80   40            01070         SUBA    #64     MAKE A CONTROL CHAR
```

```
28D3 81  0C      01080 OUT     CMPA    #$0C    INTERCEPT "FF" CHAR
28D5 26  02      01090         BNE     NOESC   BRANCH IF NOT
28D7 86  1B      01100         LDA     #$1B    MAKE AN ESCAPE CHAR
28D9 8D  4D      01110 NOESC   BSR     DATAOT  LETS GO TRANSMIT
28DB 20  BD      01120         BRA     LOOP1   GO DO IT AGAIN
                 01130 *
                 01140 *
                 01150 *
                 01160 *  DATAIN ROUTINE BRINGS IN THE DATA.  THIS ROUTINE
                 01170 *  IS THE SERVICE ROUTINE FOR THE FIRQ INTERRUPT.
                 01180 *  ALL REGISTERS ARE SAVED AND NOTHING
                 01190 *  IS REQUIRED ON ENTERING THE ROUTINE.
                 01200 *
                 01210 *
                 01220 *
28DD 34  7E      01230 DATAIN  PSHS    U,Y,X,DP,B,A    SAVE THEM
28DF 1A  50      01240         ORCC    #$50    DISABLE INTERRUPTS
28E1 8E  0078    01250         LDX     #120    GET HALF BIT WAIT VALUE
28E4 30  1F      01260 HBWAIT  LEAX    -1,X    LETS WAIT
28E6 26  FC      01270         BNE     HBWAIT  BRANCH IF NOT DONE
28E8 C6  08      01280         LDB     #8      SET BYTE COUNTER
28EA 8D  34      01290 RECDAT  BSR     WAIT    GO WAIT ONE BIT TIME
28EC 34  04      01300         PSHS    B       SAVE COUNTER
28EE F6  FF22    01310         LDB     $FF22   GET THE DATA
28F1 56          01320         RORB            PUT IT IN THE CARRY
28F2 46          01330         RORA            PUT DATA IN REGISTER A
28F3 35  04      01340         PULS    B       GET THE COUNTER
28F5 5A          01350         DECB            DECREMENT IT
28F6 26  F2      01360         BNE     RECDAT  BRANCH IF NOT DONE
28F8 46          01370         LSLA            CLEAR PARITY BIT
28F9 44          01380         LSRA
28FA 81  0A      01390         CMPA    #$0A    CHECK FOR LINE FEED
28FC 27  1A      01400         BEQ     GOODCH  YES? GO DISPLAY
28FE 81  08      01410         CMPA    #$08    CHECK FOR BACKSPACE
2900 27  16      01420         BEQ     GOODCH  YES? GO DISPLAY
2902 81  20      01430         CMPA    #$20    CHECK FOR SPACE CHAR
2904 27  12      01440         BEQ     GOODCH  YES? GO DISPLAY
2906 81  0D      01450         CMPA    #$0D    CHECK FOR CARRIAGE RETURN
2908 27  0E      01460         BEQ     GOODCH  YES? GO DISPLAY
290A 81  20      01470         CMPA    #$20    CHECK FOR CONTROL CHAR
290C 25  0C      01480         BLO     GTRDY   DON'T DISPLAY
290E 81  80      01490         CMPA    #$80    CHECK FOR HIGH RANGE
2910 25  06      01500         BLO     GOODCH  IF NOT GO DISPLAY
2912 81  40      01510         CMPA    #$40    CHECK FOR LOWER RANGE
2914 24  02      01520         BHS     GOODCH  IF NOT GO DISPLAY
2916 80  80      01530         SUBA    #$80    MAKE A DISPLAY CHAR
2918 8D  36      01540 GOODCH  BSR     SCREEN  GO PUT ON SCREEN
291A B6  FF20    01550 GTRDY   LDA     $FF20   CLEAR INTERRUPT
291D 35  7E      01560         PULS    U,Y,X,DP,B,A    RESTORE REGISTERS
291F 3B          01570         RTI             RETURN FROM INTERRUPT
2920 8E  0172    01580 WAIT    LDX     #370    LOAD ONE BIT TIME
2923 30  1F      01590 LOOP    LEAX    -1,X    WAIT
2925 26  FC      01600         BNE     LOOP    BRANCH IF NOT DONE
2927 39          01610         RTS             RETURN
                 01620 *
                 01630 *
                 01640 *
                 01650 *  DATAOT IS THE OUTPUT ROUTINE.  THE DATA MUST
                 01660 *  BE IN THE A REGISTER.  INTERRUPTS ARE
                 01670 *  MASKED.  ONLY REGISTERS A,B AND X ARE MODIFIED.
                 01680 *
                 01690 *
                 01700 *
2928 1A  50      01710 DATAOT  ORCC    #$50    DISABLE INTERRUPT
292A 5F          01720         CLRB            CLEAR COUNTER
292B F7  FF20    01730         STB     $FF20   USE ZERO TO OUTPUT START BIT
292E 8D  F0      01740         BSR     WAIT    GO WAIT ONE BIT TIME
2930 C6  08      01750         LDB     #8      LOAD COUNTER
2932 34  04      01760 LOOP3   PSHS    B       SAVE COUNTER
2934 44          01770         LSRA            PUT BIT INTO CARRY
2935 59          01780         ROLB            POSITION FOR TRANSMIT
2936 59          01790         ROLB            ONE MORE TIME
2937 F7  FF20    01800         STB     $FF20   TRANSMIT
293A 8D  E4      01810         BSR     WAIT    GO WAIT ONE BIT TIME
293C 35  04      01820         PULS    B       RESTORE COUNTER
293E 5A          01830         DECB            DECREMENT COUNTER
293F 26  F1      01840         BNE     LOOP3   BRANCH IF NOT DONE
2941 B6  FF20    01850         LDA     $FF20   GET DATA REGISTER
2944 8A  02      01860         ORA     #$02    FORCE BIT ONE HIGH
2946 B7  FF20    01870         STA     $FF20   OUTPUT STOP BIT
2949 8D  D5      01880         BSR     WAIT    GO WAIT FOR STOP BIT ONE
294B 8D  D3      01890         BSR     WAIT    WAIT FOR STOP BIT TWO
294D 1C  BF      01900         ANDCC   #$BF    ABLE INTERRUPT
294F 39          01910         RTS             RETURN
                 01920 *
                 01930 *
                 01940 *
                 01950 *  SCREEN IS USED FOR SCREEN CONTROL AND EDITING.
                 01960 *  DATA MUST BE IN THE A REGISTER.  ALL REGISTERS
                 01970 *  ARE MODIFIED AND THE NEW SCREEN LOCATION
                 01980 *  WILL BE IN H0088.
                 01990 *  NO INFORMATION IS SAVED.
                 02000 *
                 02010 *
                 02020 *
2950 10BE 0088   02030 SCREEN  LDY     >$0088  GET CURRENT SCREEN LOCATION
2954 81  08      02040         CMPA    #$08    BACK SPACE CHAR?
2956 26  0B      02050         BNE     LNFD    NO? GO AROUND
2958 B6  60      02060         LDA     #$60    GET A BLANK CHAR
295A A7  A4      02070         STA     ,Y      CLEAR CURSER
295C 86  20      02080         LDA     #$20    GET CURSER CHAR
295E A7  A2      02090         STA     ,-Y     STORE THE CURSER AND DECREMENT
```

(Continued on next page)

```
2960 16   00B6     02100          LBRA    RETURN   GO RETURN
2963 81   0A       02110 LNFD     CMPA    #$0A     LINE FEED CHAR?
2965 27   02       02120          BEQ     LNFEED   YES? BRANCH
2967 20   2B       02130          BRA     NOLNFD   GO AROUND
2969 E6   A4       02140 LNFEED   LDB     ,Y       GET CURRENT SCREEN POSITION
296B C1   20       02150          CMPB    #$20     IS IT THE CURSER?
296D 26   04       02160          BNE     NOCHNG   NO? GO AROUND
296F 86   60       02170          LDA     #$60     GET A BLANK CHAR
2971 A7   A4       02180          STA     ,Y       STORE ON SCREEN
2973 1E   02       02190 NOCHNG   EXG     D,Y      GET A BETTER REGISTER
2975 C3   0020     02200          ADDD    #$20     ADD A LINE
2978 1E   02       02210          EXG     D,Y      CHANGE IT BACK
297A 108C 0600     02220          CMPY    #$600    ARE WE OVER MAX SCREEN?
297E 24   03       02230          BHS     RESTOR   IF YES GO SCROOL
2980 16   0096     02240          LBRA    RETURN   GO TO RETURN
2983 1E   02       02250 RESTOR   EXG     D,Y      CHANGE TO BETTER REGISTER
2985 83   0020     02260          SUBD    #$20     SUBTRACT A LINE
2988 1E   02       02270          EXG     D,Y      CHANGE IT BACK
298A 30   8D 0090  02280          LEAX    FLAG,PCR       GET FLAG ADDRESS
298E C6   01       02290          LDB     #1       SET LINE FEED FLAG
2990 E7   84       02300          STB     ,X       STORE FLAG
2992 20   42       02310          BRA     CHANGE   GO MOVE THE SCREEN
2994 81   0D       02320 NOLNFD   CMPA    #$0D     CHECK FOR CARRIAGE RETURN
2996 27   02       02330          BEQ     RETN     YES? THEN BRANCH
2998 20   0C       02340          BRA     NORETN   BRANCH AROUND
299A C6   60       02350 RETN     LDB     #$60     LOAD A CLEAR CHAR
299C E7   A4       02360          STB     ,Y       STORE AT SCREEN
299E 1E   02       02370          EXG     D,Y      CHANGE REGISTERS
29A0 C4   E0       02380          ANDB    #$E0     MOVE SCREEN POSITION BACK
29A2 1E   02       02390          EXG     D,Y      CHANBE REGISTERS
29A4 20   73       02400          BRA     RETURN   GO RETURN
29A6 81   20       02410 NORETN   CMPA    #$20     SPACE CHAR?
29A8 27   02       02420          BEQ     MASK     YES? GO MAKE A DISPLAY CHAR
29AA 20   04       02430          BRA     NOMASK   GO AROUND
29AC 86   60       02440 MASK     LDA     #$60     LOAD BLANK CHAR
29AE 20   14       02450          BRA     DISP     GO DISPLAY
29B0 81   61       02460 NOMASK   CMPA    #$61     LOWER RANGE?
29B2 25   02       02470          BLO     NUMCK    BRANCH IF LOWER RANGE
29B4 80   60       02480          SUBA    #$60     MAKE A DISPLAY CHAR
29B6 81   40       02490 NUMCK    CMPA    #$40     CHECK FOR LOWER RANGE
29B8 25   02       02500          BLO     RNGCK    BRANCH IF STILL LOWER
29BA 20   08       02510          BRA     DISP     GO DISPLAY WHATS LEFT
29BC 81   21       02520 RNGCK    CMPA    #$21     CHECK FOR SPECIAL CHAR
29BE 24   02       02530          BHS     GDRNG    YES? BRANCH FOR CHANGE
29C0 20   02       02540          BRA     DISP     GO DISPLAY
29C2 8B   40       02550 GDRNG    ADDA    #$40     MAKE A DISPLAY CHAR
29C4 C6   60       02560 DISP     LDB     #$60     LOAD A BLANK CHAR
29C6 E7   A4       02570          STB     ,Y       CLEAR CURSER
29C8 A7   A0       02580          STA     ,Y+      DISPLAY AND INCREMENT
29CA 86   20       02590          LDA     #$20     LOAD THE CURSER
29CC 108C 0600     02600          CMPY    #$600    NEED TO SCROLL?
29D0 24   04       02610          BHS     CHANGE   YES? THEN BRANCH
29D2 A7   A4       02620          STA     ,Y       STORE THE CURSER
29D4 20   43       02630          BRA     RETURN   GO RETURN
29D6 8E   0400     02640 CHANGE   LDX     #$400    LOAD SCREEN START
29D9 34   20       02650          PSHS    Y        SAVE Y
29DB 108E 0600     02660          LDY     #$600    LOAD END VALUE
29DF EC   88 20    02670 INTERC   LDD     32,X     PICK UP SCROLL CHAR
29E2 10AE 88 22    02680          LDY     34,X
29E6 EE   88 24    02690          LDU     36,X
29E9 ED   81       02700          STD     ,X++     STORE THEM
29EB 10AF 81       02710          STY     ,X++
29EE EF   81       02720          STU     ,X++
29F0 BC   05DF     02730          CMPX    #$5DF    CHECK FOR END OF SCROLL
29F3 25   EA       02740          BLO     INTERC   LOOP IF NOT DONE
29F5 35   20       02750          PULS    Y        RESTORE REGISTER
29F7 30   8D 0023  02760          LEAX    FLAG,PCR
29FB A6   84       02770          LDA     ,X       CHECK FLAG FOR LINE FEED
29FD 5F            02780          CLRB             CLEAR B
29FE E7   84       02790          STB     ,X       CLEAR FLAG
2A00 81   01       02800          CMPA    #1       NEED TO FEED LINE?
2A02 27   04       02810          BEQ     LNDWN    YES? THEN BRANCH
2A04 108E 05E0     02820          LDY     #$5E0    SET NEW SCREEN VALUE
2A08 CC   6060     02830 LNDWN    LDD     #$6060   LOAD BLANK CHAR
2A0B 8E   05E0     02840          LDX     #$5E0    GET START ADDRESS
2A0E ED   81       02850 CLLN     STD     ,X++     CLEAR LINE
2A10 8C   0600     02860          CMPX    #$600    DONE?
2A13 25   F9       02870          BLO     CLLN     NO? THEN BRANCH
2A15 86   20       02880          LDA     #$20     GET A CURSER
2A17 A7   A4       02890          STA     ,Y       STORE IT ON SCREEN
2A19 10BF 0088     02900 RETURN   STY     >$0088   PUT NEW SCREEN ADDRESS IN MEM
2A1D 39            02910          RTS              RETURN
2A1E               02920 FLAG     RMB     $01      LINE FEED FLAG
2A1F      20       02930 TABLE    FCC     / MODCOMM/
          4D
          4F
          44
          43
          4F
          4D
          4D
2A27      20       02940 TBLE     FCC     /  BY WALTER CHARLTON/
          20
          42
          59
          20
          57
          41
          4C
          54
          45
          52
          20
          43
          48
          41
          52
          4C
          54
          4F
          4E
          0000     02950          END
```

A VOX Input For Your Color Computer

A discussion of subroutines that allow sound as a form of input.

All Sound Subroutines require:

TRS-80 Color Computer with cassette. Some routines require Extended BASIC.

By Sean Moyer

While waiting for a particularly long program to cassette load during one of my late night programming sessions, I began to ponder the possibility of voice control for my Color Computer. Visions of external amplifiers and analog to digital converters danced through my mind. Next to dance through my mind were visions of the price tags that accompanied these pieces of equipment. I was about to abandon this line of thought when my attention was caught by the steady hiss of the program loading in. It dawned on me that I probably had everything I needed right in front of me. I realized that in order for a program to load in from the cassette player, the input from the player had to affect any number of memory locations in the area of memory devoted to I/O. All I needed was a source of sound for the cassette player and a program to look through the area of memory devoted to I/O. I found the range of memory addresses devoted to I/O in the Color Computers Owner's Manual. I developed the program in listing 1 to look through each address, pausing momentarily at each one.

For noise, I grabbed the nearest music cassette I had on hand, put it in the Color Computer's cassette recorder and pressed play. I ran my program, sat back and waited. My wait was a short one. At memory location 65312, I noticed the content display flickering rapidly. I recorded the location, and letting the program run, I found several more locations showing identical behavior. The change on the display was binary two for the low state, three for the high.

All that remained now, was to substitute my voice for the prerecorded input. The procedure for accomplishing this varies, depending on the type of cassette you own. For the older black model, with built in microphone, simply remove the plug from the "MIC" jack on the left side of the unit. Next, open the cassette compartment and remove the cassette. Now, carefully inspect the upper lefthand corner of the compartment. There will be a small stud there. Push the stud toward the back of the unit. While holding it down, press Record and Play.

The process is the same for the more recent models with one exception. You must plug an external microphone into the "MIC" jack. With this accomplished, the recorder is now sending your amplified voice to the computer.

To test the procedure, turn up the volume on your monitor and type AUDIO ON. If the procedure was completed correctly, you should hear your amplified voice issuing from your television speaker. With the preliminary set up now complete, we are ready to discuss three subroutines that should get you sufficiently familiar with the basics of sound input to help you to develop your own sound subroutines.

The first subroutine is a voice activated binary input. When the program receives a sound, the routine is turned "ON". When the next sound is recognized, the routine is turned "OFF", and will alternate "ON" and "OFF" with each additional sound.

Specifically, the program will check memory location 65312. If this location contains a 2, the low state, it returns and checks again. If the location contains a 3, indicating the presence of sound, the program immediately prints ON. The next sequence in the routine is a FOR/NEXT loop. This delay loop debounces the switch, which is to say it allows the sound that activated the switch to end. Without this delay the voicing of a word such as ON or OFF would be sufficiently long as to turn the routine off and on several times in succession. The memory location is checked again in the manner previosly mentioned, and when a sound is recognized, the program prints OFF and goes through a debouncing loop. The program returns to

the beginning and repeates the cycle.

If the program fails to operate and you determine it has been entered correctly, reset the computer. Under special circumstances, such as the use of the AUDIO ON or PLAY commands, the contents of memory location 65312 may be other than 2 or 3. If the program gets number other than 2 or 3, it will crash. Resetting the computer should restore the program to operational order.

This is the least complex and least useful of the three routines presented here. It serves basically as a primer for the next two programs. The next two routines are more complex, and they should prove more interesting and useful as well.

The next routine we will discuss is the voice operated dictation shown in listing 3. This program is designed to record whatever you say, deleting the long periods of silence between phrases. To run this program, leave a blank cassette in the recorder before you press record and play.

The program begins by checking memory location 65312. When a sound is recognized, the recorder is activated by the MOTOR ON command. The delay loop that follows serves a somewhat different purpose than that in the VOX binary input. This loop prevents the recorder from shutting down during the short pauses between words in a sentence, or even shorter silences in the words themselves. The program begins the count from 1 to 300, and checks memory location 65312 inside the loop. If at any time a sound is recognized, the program begins the count again from 1. This way, if the pause is short, the loop reinitiates, leaving the motor on. If the pause is long, the routine will complete the loop, and the recorder will be shut down using the MOTOR OFF command. Finally, the program returns to the beginning and waits for the next sound.

If this program is used for dictation a small preliminary sound, such as a finger snap, might prove necessary for total legibility. When the first sound is sensed the motor is activated, but it takes a fraction of a second for the motor to reach proper recording speed. This often causes the first word to be lost. A finger snap would activate the motor and allow it to reach recording speed in time for the first word to be recorded. A preliminary sound would not be needed if the program were used, for instance, to monitor an empty room for activity.

The third and final program differs from the first two in that instead of a straight binary input, it uses a sampling routine to approximate an analog input. This program displays the analog approximation in PMODE 4 graphics, oscilliscope style. This is the only routine that requires Extended BASIC. In addition, because the program uses a real time graphics display, I have included the high speed poke (POKE 65495,0). If the high speed poke does not work on your system simply remove it from the program. This will slow down the routine, though the results should still be acceptable.

The routine begins with the high speed poke and then prepares the graphics screen. This is followed by a threesome of nested FOR/NEXT loops. The outer loop controls the initial vertical placement of each trace line. The next loop in controls the rate at which the trace sweeps horizontaly across the screen. The innermost loop is the heart of the program. This is the sampling routine. For every point on the horizontal sweep, memory location 65312 is sampled six times.

The routine sums all six samples, and stores the sum in variable T. It then subtracts a constant (12 in this case), so that six samples taken during a period of total silence would give T a value of zero. T is then multiplied by a constant (4) to give the graphic display proper amplitude. The grahics display comes directly after the sampling routine.

These routines will help you understand the workings of the cassette audio input port, and hopefully they will be useful to you in your program library.

You may contact Sean Moyer at, 1825 South 17th Street, Fargo, ND 51830.

Listing 1

```
10 REM PROGRAM 1
20 REM MEMORY SEARCH
30 REM BY SEAN MOYER
40 CLS
50 REM I IS ADDRESS RANGE
60 FOR I = 65280 TO 65535
70 REM J IS # OF TIMES EACH
80 REM ADDRESS IS DISPLAYED
90 FOR J = 1 TO 75
100 REM DISPLAYS ADDRESS AND
110 REM CONTENT CENTER SCREEN
120 PRINT@224,I,PEEK(I)
130 NEXT J
140 NEXT I
```

Listing 2

```
10 REM LISTING 2
20 REM VOX BINARY INPUT
30 REM BY SEAN MOYER
40 REM
50 REM CHECKS FOR SOUND
60 IF PEEK (65312) = 3 THEN 70 ELSE 60
70 PRINT"ON"
80 REM DELAY LOOP
90 REM DEBOUNCES SWITCH
100 FOR I = 1 TO 200 : NEXT I
110 REM CHECKS FOR SOUND
120 IF PEEK (65312) = 3 THEN 130 ELSE 120
130 PRINT"OFF"
140 REM DEBOUNCES SWITCH
150 FOR I = 1 TO 200 : NEXT I
160 GOTO 60
```

Listing 3

```
10 REM LISTING 3
20 REM VOX DICTATION
30 REM BY SEAN MOYER
40 REM
50 REM CHECKS FOR SOUND
60 IF PEEK (65312) = 3 THEN 80 ELSE 60
70 REM ACTIVATES RECORDER
80 MOTOR ON
90 REM DELAY LOOP/CONTINUITY
100 FOR I = 1 TO 300
110 REM IF DICTATION IN PROCESS
120 REM REINITIATES DELAY LOOP
130 IF PEEK (65312) = 3 THEN 100
140 NEXT I
150 REM DEACTIVATES RECORDER
160 MOTOR OFF
170 GOTO 50
```

Listing 4

```
10 REM LISTING 4
20 REM OSCILLISCOPE
30 REM BY SEAN MOYER
40 REM
50 REM HIGH SPEED POKE MAY NOT
60 REM WORK ON SOME SYSTEMS
70 REM REMOVE IF INOPERABLE
80 POKE 65495,0
90 REM PREPARE GRAPHICS SCREEN
100 PMODE 4,1
110 SCREEN 1,1
120 PCLS
130 REM VERTICAL PLACEMENT LOOP
140 FOR V = 26 TO 191 STEP 25
150 REM HORIZONTAL SWEEP
160 FOR H = 4 TO 255 STEP 4
170 REM SAMPLING ROUTINE
180 T=0
190 FOR I = 1 TO 6
200 T = T + PEEK (65312)
210 NEXT I
220 REM CORRECTION FOR
230 REM ZEROING AND AMPLITUDE
240 T = (T - 12) * 4
250 REM GRAPHICS DISPLAY
260 LINE (H-4, V-TD)-(H, V-T),PSET
270 TD=T
280 NEXT H
290 NEXT V
300 GOTO 50
```

MICRO

MICRO™

CoCo Bits

by John Steiner

Last month we looked at how easy it is to install drives on the CoCo. This month, lets take a look at something new that you may be interested in running with your new disk drives. The OS-9 operating system has finally been released, and is now available in local Radio Shack stores. OS-9 is one of the most powerful disk operating systems I have seen, and provides new power for CoCo, and new capabilities. In addition to OS-9, Radio Shack is selling BASIC-09, a very powerful supplement to Extended BASIC.

The OS-9 package, which retails for $69.95, comes in a gray box, and contains four operating manuals, *Getting Started With OS-9*; *OS-9 Commands*; *OS-9 Program Development*; and *OS-9 Technical Information*. OS-9 requires a 64K CoCo and at least one disk drive. See the July and August, 1983 CoCo Bits for instructions on upgrading your machine to 64K. There are two disks included in the package, an OS-9 Boot disk, and an OS-9 system master. The OS-9 boot disk allows users with 1.0 disk disk ROMs to boot OS-9. CoCo users with 1.1 disk ROM may just load the system master and type 'DOS'. The new DOS command will take care of selecting 64K mode and other details.

The boot disk has one other utility of value to all CoCo disk users, a disk timing utility. This program reads and displays the RPM of the selected drive. The *Getting Started...* manual lists accepted speeds, however an addendum letter corrects those speeds to a range of acceptable RPM from 295.5 to 304.5 RPM. If your drive is out of time, Radio Shack recommends you send it in to be adjusted, however the TEC drive included with the RS disk system is easy to adjust yourself.

All you have to do is shut off the computer, unplug the drive from the AC outlet, remove the screws holding the case and lift the case off. On the left side of the drive (facing front), you will find the motor drive belt and pulleys. Just below these, there is a small circuit board with a small adjustment potentiometer. Plug in the drive unit, install and run the disk timing program and adjust the potentiometer with a small screwdriver until the drive speed is within range. When done, turn off the computer, unplug the drive and reinstall the case.

Now that we are done with that little diversion, back to OS-9. Other included utilities include an editor and assembler for writing your own OS-9 software. For those who would like to know how the DOS interfaces with your CoCo, the operating system resides between any applications programs and languages you may run, and the computer. All programs and languages are designed to use OS-9 to access the disk, printer, terminal, keyboard and video display.

Who needs OS-9? Anyone who wants to run applications programs written using the operating system, and anyone who might be interested in running other languages besides BASIC. By the time you read this, you should be able to buy PASCAL, C, and COBOL configured to run on CoCo. If you write your own machine language software, you will like the thorough documentation of the operating system, and the ease with which you can implement and access disk files, and other CoCo accessories without having to write your own device drivers or trying to decipher the ROM.

OS-9 is in charge of managing the disk directory, and keeping tabs on how the computer can manage its time and system resources. OS-9 creates one of the most sophisticated disk directories in the computer industry in that it allows directories within a directory. For example, you can make a directory of accounting files called ACCOUNTS, and within the directory you might have files called ledger, receivables, payables, etc. For that matter, you could have a sub directory in the ACCOUNTS directory that could have its own set of files. If you enter DIR at the OS-9 prompt, you will see a list of directories and files stored under the main operating system directory. Entering DIR /D0/ACCOUNTS will list all the files and sub directories stored in the ACCOUNTS directory. To keep files and directories separate, OS-9 users have developed the convention of using upper case characters when naming characters and lower case characters when naming files. Eight letter filenames are a thing of the past and you are allowed from 1 to 29 characters.

The directory system builds files in a tree or hierarchical format and you move from directory to directory with the CHD (change data directory) or CHX (change execution directory) commands. CHD /D0/ACCOUNTS will move you to the ACCOUNTS directory and all commands that contain any default options will default to the ACCOUNTS directory. The whole process is a lot easier to understand than explain, and the manuals have several examples that allow you to understand how to create and delete directories.

In addition to the multiple level filing system, OS-9 is multiuser and multitasking. In the Color Computer implementation, you can configure the RS-232 port as a terminal and someone can use the accounting package from a terminal while you are running customer files from the main CoCo keyboard. A security system can be implemented that allows users to have private and public files, and files can be read or write protected. You have the capacity of private passwords for up to 65,535 users. Most of us don't even know that many people.

Multitasking allows the user to run more than one task at a time. For example, you might ask OS-9 to list a file using the command LIST /D1/customer /P &. LIST is a program that will display the output of a file on the video display. In this case, the file "customer". Options include /D1 for drive one, /P redirects output from the video display to the printer and the & (ampersand) tells OS-9 to do this as a background task. This means that the prompt will return to the display, and keyboard control will be restored so that you may continue with other tasks. OS-9 will share time between the printing task you just gave it and any other tasks you decide to give it. The only limit to the number of tasks

you ask the computer to do at one time is the available memory space.

The computer will share tasks in a priority basis, and you may assign a higher priority for jobs you want done more quickly. OS-9 uses a hardware clock to divide time into slices, and allot more time slices to tasks that have a higher priority. By the way, when you sign on to OS-9, you can enter the time and date, and the time will remain accurate, even with disk I/O as the software does not allow interruption of the time clock for disk I/O. You can use the clock in program output for any purpose desired.

One of the major advantages of OS-9 is its device independent I/O system. Only one I/O routine is needed with several drivers written to support the devices being used. OS-9 normally defaults to input from the keyboard and output to the video display, however it is easy to redirect input and output as required. Devices that are supported in the CoCo implementation are:

Drives 0 to 3
Keyboard
Video display
Serial printer
RS-232 terminal
Joysticks

As an example, you might redirect input from the keyboard to drive 1, and output from the display to the printer.

OS-9 is powerful, and complex, and I have a big project to become proficient in its use during the cold North Dakota winter. This overview will hopefully give you some idea of the new found power in the Color Computer. You don't have to hang your head anymore when someone asks "Why did you buy that "toy" computer.

Over the next few months we will take other looks at OS-9 and its features. Next month we will look at BASIC09, and see what features and advantages you get for your hundred dollars. We will also take a look at two more CoCo word processors, Nelson's VIP Writer, and Elite*Word. MICRO

You may contact John at 508 Fourth Ave., N.W., Riverside, ND 58078.

SPECTRUM

32K RAM Button	$ 2.99
NANOS Reference Card	$ 3.99
64K RAM Button	$ 4.99
Coco Editor Assembler	$ 6.95
Coco Tech Manual	$ 7.95
16K RAM Chips	$ 9.95
Coco Secrets Revealed Book	$ 14.95
LED On/Off Indicator	$ 14.95
Coco Light Pen	$ 19.95
ATARI Joystick Interface	$ 19.95
Video Interface Kit	$ 24.95
16K-32K Upgrade Kit	$ 25.95
6883 SAM Chip	$ 29.95
6809E CPU Chip	$ 29.95
Basic ROM 1.1	$ 36.00
64K RAM Chips	$ 49.95
MARK DATA Keyboard	$ 69.95
BOTEK Printer Interface	$ 69.95
Extended Basic ROM	$ 84.00
Disk Controller	$139.95
COLOR COMPUTERS	$ CALL

Call or Write for FREE Catalog
SPECTRUM PROJECTS
93-15 86th Dr Woodhaven, New York 11421
Add Sales Tax & $3.00 for S/H
●● Dealer/Club Inquires Invited ●●

212 441-2807

It pays to write for MICRO!

- ☐ **Win the esteem of your colleagues ·**
- ☐ **Gain recognition as a computerist ·**
- ☐ **See your programs and ideas in print ·**
- ☐ **Feel satisfaction from helping others ·**
- ☐ **Earn money for your work.**

MICRO wants to continue to provide substantial material for the **Serious Computerist**. This must come from you - the readers of **MICRO** who are actively invloved with understanding the inner workings of microcomputers, who experiment to find better ways of using microcomputers, who are developing new techniques, who are programming in assembler, FORTH, Pascal and other languages in addition to BASIC.

Writing for **MICRO** is easier than you think. If you can provide the programs and the basic information about how they work, our staff can help turn this 'rough' material into a polished article. Remember, in this business, everyone knows something of value, and nobody knows everything.

Request the **Writer's Guide** which covers many of the fundamentals of writing for **MICRO**.

Product Name: **Disk interface/ROM pack extender**
Equip. req'd: TRS-80 Color Computer Disk system
Price: $29.95 + $3.00 shipping
Manufacturer: Spectrum Projects
93-15 86th Drive
Woodhaven, NY 11421

Description: The Spectrum Projects disk interface and extender cable can be a useful accessory for the Color Computer. It allows the disk or program ROM pack to be located up to three feet away from the side of the computer. This high quality cable has a gold plated connector on each end that mates with the connectors on the computer and ROM packs.

Pluses: Makes positioning of drives more convenient at the desk, allows the computer to have only a ribbon cable extending from the computer, rather than a large ROM pack.

Minuses: Unfortunately, it may not work with all drives. My MPI works correctly with the cable but my Tandy unit has problems. If I put drive one as drive zero, then the cable seems to work. It is possible that my drive needs adjustment for the longer length of cable.

Documentation: None needed or provided. The cable is marked TOP on both sides, care must be taken that the cable is inserted correctly.

Skill level required: The cable is easy to install by even a novice.

Reviewer: John Steiner

Product Name: **The Stripper**
Equip. req'd: TRS-80 Color Computer
Price: $7.95
Manufacturer: Eigen Systems
P.O. Box 10234
Austin, TX 78766

Description: *The Stripper* is a programmers utility for the Color Computer that will compress or "pack" a BASIC program. Stripper has three options, remove remarks, pack lines, and remove spaces. Any one or all three options may be exercised.

Pluses: The packed program will take up less memory, and may operate slightly faster. The utility normally loads into graphics memory, thus freeing all other memory for the packing operation, and your BASIC program. One of the lowest cost utilities I've seen, and well worth the money.

Minuses: You cannot edit, or otherwise change a line packed program. The program must be saved in an unpacked format, should you decide to modify or edit it.

Documentation: Four pages of notes describe the operation of the program thoroughly.

Skill level required: No advanced skill level is required to use the program.

Reviewer: John Steiner

Product Name: **CCP-1 Serial-Parallel Interface**
Equip. req'd: TRS-80 Color Computer
Price: $69.96 + shipping
Manufacturer: Botek Instruments
4949 Hampshire
Utica, MI 48087

Description: Botek Instruments serial-parallel interface provides the TRS-80 Color Computer and TDP-100 with a standard Centronics style parallel interface. The interface, housed in a 3 in. by 6 in. mini-box, is easily connected to the CoCo RS-232 port. A parallel cable is provided that connects the interface to the printer. A third cable leads to a small AC power supply. Baud rates of 300 to 9600 baud are switch selectable on the front of the interface. POKE 150,1 will allow CoCo to access the printer at 9600 baud. The unit can be configured to ROM 1.0 or ROM 1.1 by just changing a jumper.

Pluses: The interface is easily installed, and requires little attention, except to change baud rates. If the printer you have is capable of +5 volts on pin 18 of the printer connector, you may remove the AC supply and power the interface from your printer. This option is not available on my Epson printer, so I could not test it.

Minuses: None noted.

Documentation: A single page instruction sheet describes hook-up and operation of the unit. In addition, a separate sheet contains a complete interface schematic.

Skill level required: The unit should be easy to install for anyone familiar with computer and printer installations.

Reviewer: John Steiner

Product Name: **BASIC Aid**
Equip. req'd: TRS-80 Color Computer
Price: $34.95

Manufacturer: Eigen Systems
P.O. Box 10234
Austin, TX 78766

Description: *BASIC Aid* is a ROM based programmers utility for the Color Computer. Commands may be entered using two keystrokes, and keys may be redefined as required. A plastic overlay is included to remind you of the key codes. Automatic line numbering is supported, and lines may be moved from one part of a program to another. A MERGE utility allows two BASIC programs to be combined.

Pluses: *BASIC Aid* works with all CoCo's, 4K on up. ROM pack construction requires only 227 bytes of RAM to hold the key define table. An auto space command prints a space after the defined key for maximum readability. Auto space can be shut off should you require memory conservation. Programs to be merged may be binary, and do not have to be stored in ASCII format unlike many utilities. BASIC Aid can be turned off and on from the keyboard if desired.

Minuses: *BASIC Aid* automatically allocates the top 227 bytes of available RAM for its key define table, which is not relocatable. As a result, you will have to relocate any machine-language routines that must be used with your program, assuming they are relocatable. In addition, all CLEAR commands must be set to protect the top 227 bytes, or you may accidentally overwrite the table.

Documentation: The instruction book is excellent, easy to understand, and well written. The only conflict I could find was in the MOVE utility. The manual asks for line numbers in the format 100, but on my machine, 100 turned into 10000 until I entered the numbers with leading zeroes, as 00100.

Skill level required: The program is easy to use, and makes programming a much easier task. No extra skills are required of the programmer.

Reviewer: John Steiner

Product Name: **CCEAD Color Computer Editor/ Assembler/Debugger**
Equip. req'd: TRS-80 Color Computer
Price: $6.95 + $3.00 shipping
Manufacturer: Eigen Systems
PO Box 10234
Austin, TX 78766

Description: This is a BASIC program that provides the user with a machine-language software development system. You receive two versions: a 12K+ commented source listing and a stripped 7600-byte running version. It assembles and stores code in the unused CoCo graphics pages.

(Continued on next page)

PERRY PERIPHERALS

COMPLETE PRODUCT LINES FOR
D.C. Hayes • HDE • Microsoft • Nashua • Okidata

APPLE • FRANKLIN • IBM-PC/XT

We carry many products at competitive prices to expand these fine computers.
Request Catalog No. AFP*

Dealer Inquiries Invited

HDE • AIM • SYM • KIM

HDE hardware and software for ASK, DLC, OMNI-65 systems
KIM replacement modules (1K — 4K) and keyboards
AIM-65 (1K — 4K) and accessories. Request Catalog No. TASK*

COMPUTER REPAIR SERVICE

Professional Workmanship
Guaranteed Repairs
Quick Turn-around

Apple • Franklin • Atari
PET • HDE
AIM • SYM • KIM

Perry Peripherals

*Catalogs
AFP $2.00 TASK $1.50
Catalog price refunded with first order

Repair Center
6 Brookhaven Drive
Rocky Point, NY 11778

Orders
P.O. Box 924
Miller Place NY 11764
(516) 744-6462
9AM—5PM Weekdays

Pluses: The program is inexpensive and easy to use. It allows you to "get your feet wet" in machine-language programming for little cost. The BASIC program is easily modified to the user's preference.

Minuses: The program is slow in assembly and allows only five directives; ORG, RMB, FCB, FDB, and EQU. No math is allowed in labels.

Documentation: A ten-page manual is provided to demonstrate the operation and use of CCEAD. Two sample programs are provided that assist in learning to use the software.

Skill level required: This program requires some knowledge of machine-language programming.

Reviewer: John Steiner

Product Name: **SHRINK**
Equip. req'd: Color Computer with 16K memory
Price: $15.00 tape or disk
Manufacturer: Star Kits
P. O. Box 209-N
Mt. Kisco, NY 10549

Description: *SHRINK* is another implementation of Eliza, the computerized psychologist. It is fun to use, but it is rather easily "fooled" into making somewhat irrelevant comments. It will run indefinitely and is a cute demo for parties, etc.

Pluses: *SHRINK* is a change of pace: you can relax while experimenting — an unusual use for a computer. Written in assembly language, it is somewhat faster than many other similar programs.

Minuses: A serious experimenter might wish for a more sophisticated implementation of this concept.

Documentation: A single page gives adequate instructions; once started, all you need to know is how to stop — simply type "shut up"!

Skill level required: None.

Reviewer: Ralph Tenny

MICRO™

SCIENTIFIC SOFTWARE ASSOCIATES, LTD.

Q-card

Questionnaire Analysis Software

- Microcomputer based
  Avoid the expense of contract services -- do everything in-house on your own Apple II+ microcomputer.
- Easy data entry
  Avoid time consuming keypunching. Uses respondent-marked cards entered with an Optical Mark Reader (keyboard entry also possible).
- Comprehensive data analysis
  Sort on any variable(s), tally all responses, conduct cross tabs, correlations, linear regression, frequency distributions, and more.
- Complete editing capabilities
  Weight items, derive composites, add or delete items, and more.
- Easy-to-use
  Programs are user friendly, menu driven, and interactive. No special computer expertise is required.

Call or send for more information today.

SCIENTIFIC SOFTWARE ASSOCIATES, LTD.

BOX 208 · WAUSAU, WI. 54401

TELEPHONE: (715) 845-2066

Apple II+ is a registered trademark of Apple Computer, Inc.

NOW — FULL POWER ASSEMBLY LANGUAGE!!

GET THE VERSATILITY OF INDEPENDENTLY ASSEMBLED RELOCATABLE SUBROUTINES WITH A

RELOCATING LINKING LOADER

* For use with APPLE DOS TOOLKIT ASSEMBLER REL files
* Relocate and link up to 255 REL files
* Convenient specification of relocation parameters (output file, start address, REL file names, . . .) in text control file
* Comprehensive error reporting and recovery
* Memory map has module and entry point addresses to aid debugging
* Complete with user manual and disk example
* FAST — ten 256 byte REL files done in 30 SECS

RELOCATING LINKING LOADER W/MANUAL.....$49.95
MANUAL.....$ 4.50

Also — FAST-CRYPT — Available for APPLE and IBM
Encrypt/decript any file with FAST-CRYPT
An efficient assembly language implementation of the DES (Data Encryption Standard) algorithm
FAST-CRYPT W/MANUAL.....$34.95

KIWI SOFTWARE CO.
P. O. BOX 218 m
Plainsboro, N.J. 08536

System requirements:
APPLE II, II+, IIe (or compatable) DOS 3.3 W/48K
IBM-PC, XT (or compatable) PC, MS-DOS W/64K
IBM tm IBM CORP, APPLE tm APPLE COMPUTER, INC.
MS-DOS tm MICROSOFT CORP.

KIWI SOFTWARE does not copy protect its products
send check or money order (N.J. residents add 6% sales tax)

Apple To Smartmodem

By John Kelty

Adapted for Apple by Phil Daley

This is an adaptation of my program for the Color Computer. The hardware requirements are the Hayes Smart Modem and an RS-232 communications interface. This program was specifically written with the CPS Multifunction card in slot 1 with the serial port assigned to slot 2 (the standard slot for communications). This program should be easily adaptable to other interfaces. See your interface manual for any necessary changes.

This fantastic modem can be programmed from Applesoft BASIC.

The interface cable is a standard communications cable — DB-25 to DB-25, with pin-for-pin connections. You need 12-conductor cable and I have used as long as about 25 feet without any problems. It is convenient however, to have the Smartmodem near your computer setup, then you can hear the dialup and the send and receive carrier tones when you begin communicating. There are many programming options you can choose with the Hayes Smartmodem, and these are extremely well documented in the excellent Owner's Manual. (In case you have not guessed, I consider the Hayes Smartmodem my best computing purchase to date!).

The BASIC program is menu driven for ease of operation. If you BREAK and then RUN the program again, the Smartmodem functions are cleared ("AT Z" is sent to the modem). To write to the Hayes, merely access the slot (with PR#2), and PRINT your commands. Now you can talk to your modem just like it was a printer with PRINT "your modem command". To get back into the normal Apple mode use "PR#0".

Now you can use a communications package such as DATA CAPTURE or a similar program to communicate with bulletin boards or data bases. The program includes a command to enable terminal mode on the CPS card, which will allow a dumb terminal communications setup.

I know little to nothing of amateur radio operations, but with a few lines of BASIC and possibly some hardware, the Hayes Smartmodem can be made to send morse code. Auxiliary relay contacts are provided in the modem to key the transmitter automatically. It cannot however, be made to directly receive morse code. Diagrams describing the modem to radio interface are found in the Smartmodem User Manual Appendix H. Obviously some nice commercial terminal packages exist for the Apple, but this program has made me happy for some time now. With the Hayes Smartmodem and the Apple Computer, I am sure you will also enjoy computer communications.

Listing 1

```
100  REM SMART300
110  REM KELTY ENGINEERING
120  REM WRITTEN BY JOHN R. KELTY
130  REM LINCOLN, NEBRASKA  68505
140  REM
150  REM USE THIS PROGRAM TO CONNECT
160  REM AND PROGRAM THE HAYES
170  REM SMARTMODEM TO THE
180  REM APPLE COMPUTER.
190  REM
200  REM THE CONNECTOR CABLE SHOULD
210  REM CONNECT TO THE RS-232 PORT
220  REM IN THE FOLLOWIMG WAY:
230  REM PIN FOR PIN CONNECTION
260  REM
270  REM THE 300 BAUD HAYES MODEM
280  REM MAY BE USED WITH VIDEOTEX
290  REM OR ANY OTHER TERMINAL
300  REM PACKAGE THAT ALLOWS THE
310  REM HAYES TO RUN AT 300 BAUD.
320  REM
330  REM THE HAYES WILL OPERATE
340  REM LOCALLY AT HIGHER RATES
350  REM BUT NO HIGHER THAN 300
360  REM WHILE ON LINE.
370  REM
380  REM SEE YOUR USER'S MANUAL
390  REM FROM HAYES FOR DETAILS.
400  REM
410  REM ******************************
420  DIM M$(5),SN$(9):D$ =  CHR$ (13) +  CHR$
     (4)
430  HOME : PRINT : PRINT "    ***** SMART M
     ODEM *****"
440  PRINT "     BY: KELTY ENGINEERING"
450  PRINT "    ************************"
460  PRINT : PRINT "CONNECT CABLES TO MODEM,
     "
470  PRINT "COMPUTER AND PHONE COMPANY."
480  PRINT : PRINT : PRINT "(BREAK AND LIST
     THIS PROGRAM TO"
490  PRINT "SEE CABLE DETAILS)."
500  PRINT : PRINT : INPUT "PRESS ENTER WHEN
      READY";A$
510  HOME : PRINT
520  PRINT D$"PR#2": PRINT "AT Z": REM  CLEA
     R MODEM
530  PRINT D$"PR#0"
540  REM ******************************
550  REM MENU
560 M$(1) = "AUTO DIALING"
570 M$(2) = "AUTO ANSWERING"
580  FOR S = 1 TO 2
590  PRINT S"  "M$(S)
600  NEXT S
610  PRINT : PRINT : PRINT "SELECT ONE OF TH
     E ABOVE"
620  GET A$
630  ON  VAL (A$) GOSUB 650,1210
640  GOTO 430
650  REM ******************************
660  REM AUTO DIALING ROUTINE
670  REM ******************************
680  REM
690  HOME : PRINT : PRINT M$(1)
700  PRINT : PRINT : PRINT "1 DIAL A SAVED N
     UMBER"
710  PRINT "2 DIAL A NEW NUMBER"
720  GET A$
730  ON  VAL (A$) GOTO 750,950
740  GOTO 690
750  REM ************************
760 HM = 2: REM  HOW MANY NUMBERS
770  REM TO CHANGE THIS LIST, MAKE HM=HOW
780  REM MANY # YOU WANT TO LIST (MAX OF 9).

790  REM THEN INSERT LINES SIMILAR TO THOSE
800  REM BELOW WITH YOUR NUMBERS AND NAMES.
810  HOME : PRINT "NUMBERS YOU CAN EASILY CA
     LL": PRINT
820 SN$(1) = "1   XXX-XXXX          A FRIEND
     "
830 SN$(2) = "2   XXX-XXX-XXXX      A BBS
     "
```

PUT THE FULL POWER OF YOUR VIC-20 AT YOUR COMMAND!

Order your copy of MICRO's newest book...

MASTERING YOUR VIC-20

with eight BASIC projects

Now you can do more with your VIC-20. This new book and the eight projects and programs it contains can teach you how to master VIC BASIC programming. Each chapter concentrates on a particular aspect of VIC BASIC, and each program is accompanied by discovery-oriented, tutorial text. Clear directions that will quickly have you writing programs, modifying them and adding features all on your own. And to help you master your VIC-20 even faster, all eight programs in the book are already keyed in on the accompanying cassette.

You'll Receive:

✦ **MICRO Calc...a miniature spread-sheet program that makes complex, repetitive calculations a breeze.**

✦ **MASTER...a guessing game that teaches programming with random numbers and flags.**

✦ **VIC Clock...to teach you ON..GOSUB function and character graphics.**

✦ **BREAK-UP...a popular game that also teaches how animation is achieved with PEEKS and POKES to screen memory.**

✦ ***Plus...*** **music programming, string manipulation, sorting demonstrations, and more.**

Use this coupon or the postage paid card in this issue to order.

MICRO Books P.O. Box 6502, Chelmsford, MA 01824

☐ YES, please rush _____ copies of MASTERING YOUR VIC-20 (w/cassette), at only $19.95 per copy (plus $2.00 s/h, MA res. add 5% sales tax).

My payment of $_____ is enclosed. I'm paying by ☐ Check ☐ MO ☐ VISA ☐ MC

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

CREDIT CARD # _____ EXP. DATE _____

Allow 6-8 weeks for delivery.

Each Program Worth the Price of the Book!

Order your copy of MASTERING YOUR VIC-20 Today!

COMPUTER MAIL ORDER

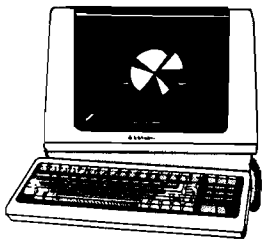
TERMINALS

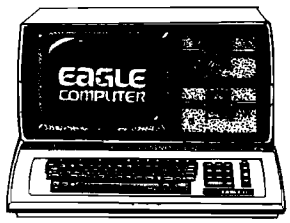
910	$519.00
910 Plus	$599.00
914	$569.00
924	$689.00
925	$739.00
950	$929.00
970	$1039.00

COMPUTERS

Teleport Portable	CALL
800A	$1099.00
802	$2699.00
803	$1949.00
802H	$4695.00
806/20	$4999.00
816/40	$9199.00
1602	$3399.00
1603	CALL

TeleVideo

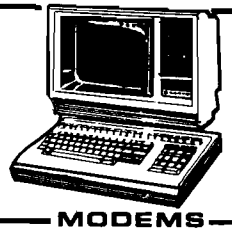




Spirit XL Portable	$3999.00
IIE-1	$1369.00
IIE-2	$1649.00
IIE-3	$2399.00
IIE-4	$3199.00
PC-E	$1579.00
PC-1	$2399.00
PC-2	$2799.00
PC-XL	$3599.00
1620	$3599.00
1630	$5499.00
1640	$6499.00
Cyma Software	CALL

SANYO

MBC-550 PC	CALL
MBC-555 PC	CALL
MBC 1100	$1599.00
MBC 1150	$2099.00
MBC 1200	$1999.00
MBC 1250	$2399.00
FDD 3200-320K Drive	$399.00
FDD 6400-64K Drive	$499.00
PR 5500 Printer	$699.00



MONITORS

AMDEK

300 Green	$149.00
300 Amber	$159.00
310 Amber	$169.00
Color 1	$279.00
Color 1 Plus	$299.00
Color 2	$399.00
Color 2 Plus	$419.00
Color 3	$349.00
Color 4	$699.00

BMC

12" Green	$79.99
12" Green HI-RES	$119.99
9191-13" Color	$249.99

GORILLA

12" Green	$88.99
12" Amber	$95.99

NEC

JB 1260 Green	$109.99
JB 1201 Green	$149.99
JB 1205 Amber	$159.99
JC 1215 Color	$299.99
JC 1216 RGB	$429.99

QUADRAM

Quadchrome	$499.99

TAXAN

12" Green	$119.00
12" Amber	$129.00
Taxan 1 RGB	$279.00

USI

Pi 1, 9" Green	$99.99
Pi 2, 12" Green	$119.99
Pi 3, 12" Amber	$149.99
Pi 4, 9" Amber	$139.99
1400 Color	$269.99

ZENITH

ZVM 122 Amber	$99.99
ZVM 123 Green	$89.99
ZVM 135 Color/RGB	$469.99

MODEMS

ANCHOR

Mark I (RS-232)	$79.00
Mark II (Atari)	$79.00
Mark III (TI-99)	$109.00
Mark IV (CBM/PET)	$125.00
Mark V (Osborne)	$95.00
Mark VI (IBM-PC)	$169.00
Mark VII (Auto Ans/Auto Dial)	$119.00
Mark XII (1200 Baud)	$299.00
TRS-80 Color Computer	$99.00
9 Volt Power Supply	$9.00

HAYES

Smartmodem 300	$219.00
Smartmodem 1200	$509.00
Smartmodem 1200B	$459.00
Micromodem II	$265.00
Micromodem II Plus	$299.00
Micromodem IIE	$269.00
Micromodem 100	$299.00
Smart Com II	$89.00
Chronograph	$199.00

NOVATION

J-Cat	$99.99
SmartCat 103	$179.00
SmartCat 103/212	$399.00
AutoCat	$219.00
212 AutoCat	$549.00
Apple Cat II	$249.00
212 Apple Cat	$569.00
Apple Cat 212 Upgrade	$309.00
Cat	$139.99
D-Cat	$149.00
PC-Cat	$339.00

ZENITH

ZT-1	$369.00

APPLE INTERFACE CARDS & BUFFERS

Choose from PKASO, Orange Micro, MPC, MicroMax, Tymac, Quadram & Practical Peripherals ..... CALL

IBM

NEC 3550 ..... $1699.00

PERCOM/TANOON DRIVES

5¼" 320K Floppy	$249.00
5 Meg Hard w/Controller	$1399.00
10 Meg Hard w/Controller	$1699.00
15 Meg Hard w/Controller	$2095.00
20 Meg Hard w/Controller	$2399.00

AMOEK

310A Amber Monitor	$169.00
DXY 100 Plotter	$599.00
Color II	$399.00

AST RESEARCH, INC.

Six Pak Plus...from	$279.00
Combo Plus II...from	$279.00
Mega Plus...from	$309.00
I/O Plus II...from	$139.00

QUADRAM

Quadlink	$549.00
Quadboard...as low as	$289.00
Quad 512 Plus...as low as	$249.00
Quadcolor...as low as	$219.00
Chronograph	$89.00
Parallel Interface Board	$89.00
64K RAM Chips Kit	$59.00

MICRO PRO

WordStar/Mail Merge	$369.00
InfoStar	$299.00
SpellStar	$159.00
CalcStar	$99.00

MICROSTUF

Crosstalk	$105.00

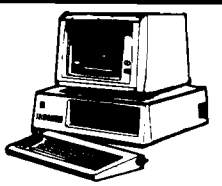
MICROSOFT

Multiplan	$159.00

pfs

	APPLE	IBM
Write:	79.00	89.00
Graph:	79.00	89.00
Report:	79.00	79.00
File:	79.00	89.00
Solutions*: as low as	16.00	16.00

*Call On Titles



ASHTON-TATE

dBASE II	$399.00

IUS

EasyWriter II	$189.00
EasySpeller	$119.00
EasyFiler	$229.00

CONTINENTAL SOFTWARE

1st Class Mail/Form Letter	$79.00
The Home Accountant Plus	$88.00

SYNAPSE

File Manager	$89.00

LOTUS

1-2-3	$319.00

PROFESSIONAL SOFTWARE

PC Plus/The Boss	$379.00

VISICORP

	IBM	APPLE
VisiCalc		159.00
VisiCalc 4	159.00	
VisiCalc-Advanced		269.00
VisiWord/Spell	249.00	
Visitrend/Plot	199.00	199.00
VisiLink		169.00
VisiFile	199.00	169.00
VisiSchedule	199.00	199.00
Visidex		159.00
VisiPlot		135.00
VisiTerm		75.00
Desktop Plan	199.00	169.00
Bus. Forecast Model	75.00	75.00
Stretch Calc	75.00	75.00
VisiTutor Calc	59.00	59.00
VisiTutor-Advanced	75.00	75.00
VisiTutor Word	59.00	59.00
Vision Calc	249.00	
Vision Graph	129.00	
Vision Mouse	159.00	
Vision Host	319.00	

PRINTERS

AXIOM

AT-100 Atari Interface	$249.00
CD-100 CBM 64/VIC 20	$249.00
GP-100 Parallel Interface	$199.00

BMC

401 Letter Quality	$589.00
BX-80 Dot Matrix	$269.00

CENTRONICS

122 Parallel	$499.00
739-1 Parallel	$399.00
739-3 Serial	$449.00

C.ITOH

Gorilla Banana	$209.00
Prowriter 8510P	$379.00
Prowriter 1550P	$679.00
A10 (18 cps)	569.00
F10-40	$999.00
F10-55	$1499.00

COMREX

ComWriter II Letter Quality	$549.00

DIABLO

620 Letter Quality	$949.00
630 Letter Quality	$1749.00

DAISYWRITER

2000	$999.00
2500	CALL
Tractor Feed	$109.00

EPSON

MX-80FT, MX-100, RX-80, RX-80FT, FX-80, FX-100	CALL

IDS

Prism 80...For Configurations	CALL
Prism 32...For Configurations	CALL

MANNESMAN TALLY

160L	$589.00
180L	$799.00
Spirit 80	$309.00

NEC

8023 Dot Matrix	$379.00
8025 Dot Matrix	$669.00
2050 Letter Quality	$999.00
3510 Serial/Letter Quality	$1449.00
3530 Parallel/Letter Quality	$1499.00
3550 IBM/Letter Quality	$1699.00
7710/7730 Serial/Parallel	$1949.00

OKIDATA

82,83,84,92,93,2350,2410	CALL

SMITH CORONA

TP-2	$449.00
Tractor Feed	$119.00

SILVER REED

500 Letter Quality	$469.00
550 Letter Quality	$699.00

STAR

Gemini 10X	$299.00
Gemini P15	$449.00
Delta 10	$559.00
Serial Board	$75.00

TOSHIBA ..... CALL

TRANSTAR ..... CALL

PRINTER CABLES

Available for Atari, Commodore, IBM, Apple, Epson, Kaypro, Televideo, Franklin, Eagle, Sanyo, Osborne, NEC, Zenith and many others. We supply all your computer needs!

PAPER SUPPLIES

1000 shts. 8½x11 Tractor Paper	19.99
1000 shts. 14½x11 Tractor Paper	$24.99
1 or 2" Address Labels	$9.99

=WEST= 800·648·3311

In NV call (702)588-5654. Dept 0115
Order Status Number: 588-5654
P.O. Box 6689, Stateline, NV 89449

=CANADA= 800·268·4559

In Toronto call (416-828-0866. Dept 0115
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

=EAST= 800·233·8950

In PA call (717)327-9575. Dept 0115
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St., Williamsport, PA 17701

MasterCard

No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the UPS continental United States with no waiting period for certified checks or money orders. Add 3% (minimum $5.00) shipping and handling on all C.O.D. and credit card orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability and price change. We stock manufacturer's and third party software for most all computers on the market. Call today for our catalog.

COMPUTER MAIL ORDER

FRANKLIN

ACE 1000 Color Computer ....CALL
ACE Family Pack System .....CALL
ACE PRO PLUS System .....CALL
ACE 1200 Office Mgmt. System ...CALL
"NOT IIE EXPENSIVE"

APPLE/FRANKLIN DISK DRIVES

MICRO-SCI

A2 .....$219.00
A40 .....$299.00
A70 .....$319.00
C2 Controller $79.00
C47 Controller $89.00

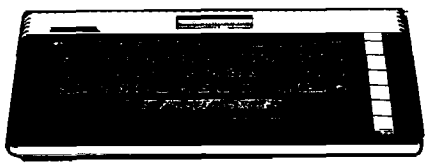
RANA

Elite 1 .....$279.00
Elite 2 .....$389.00
Elite 3 .....$569.00

APPLE IIe STARTER PACK
64K Apple IIe, Disk Drive & Controller, 80 Column Card, Monitor II & DOS 3.3
COMPLETE $1499.00

HOME COMPUTERS

ATARI



600XL..... $189
800XL..... $299

1200XL .... CALL
1400XL .... CALL

1010 Recorder $74.00
1020 Color Printer $249.00
1025 Dot Matrix Printer.....$449.00
1027 Letter Quality $299.00
1030 Direct Connect Modem...CALL
1050 Disk Drive $379.00
CX30 Paddle $12.00
CX40 Joystick.....each...$8.00
CX77 Touch Tablet $64.00
CX80 Trak Ball $48.00
CX85 Keypad $105.00
4003 Assorted Education....$47.00
4011 Star Raiders $33.00
4012 Missile Command $29.00
4013 Asteroids $29.00
5049 VisiCalc $159.00
7097 Logo $79.00
7101 Entertainer $69.00
7102 Arcade Champ $75.00
8026 Dig Dug $33.00
8030 E.T. Phone Home $33.00
8031 Donkey Kong $39.00
8036 Atari Writer $79.00
8040 Donkey Kong, Jr. $39.00
8043 Ms. PacMan $39.00
8044 Joust $39.00

commodore

CBM 8032 $599

CBM 4032.....$599.00
CBM 8096 .....$869.00
CBM 9000.....$999.00
B128-80.....$769.00
CBM 64K Memory Board....$269.00
8032 to 9000 Upgrade $269.00
2031 LP Disk Drive $299.00
8050 Disk Drive.....$949.00
8250 Disk Drive $1199.00
4023 Printer.....$379.00
8023 Printer.....$569.00
6400 Printer $1399.00
Z-RAM $499.00
Silicon Office $699.00
The Manager.....$199.00
Soft ROM $125.00
VisiCalc $159.00

CBM 64 ..... $199
VIC 20 ..... CALL

C1541 Disk Drive $249.00
C1530 Datasette $69.00
C1520 Color Printer/Plotter...$169.00
C1525 Dot Matrix/Parallel...$219.00
C1526 Dot Matrix/Serial $279.00
C1702 Color Monitor $249.00
C1311 Joystick $4.99
C1312 Paddles.....$11.99
C1600 VIC Modem $89.00
C1650 Auto Modem.....$89.00
Logo 64.....$49.00
Pilot 64 .....$39.00
Simon's Basic $19.00
Word Pro 64 Plus.....$59.00
Parallel Printer Interface.....$69.00
Calc Result 64 .....$129.00
Codewriter 64 .....$75.00
Quick Brown Fox.....$49.00
Word Pro 64 Plus.....$59.00

PROFESSIONAL SOFTWARE

Word Pro 2 Plus.....$159.00
Word Pro 3 Plus.....$189.00
Word Pro 4 Plus/5 Plus...each...$279.00
InfoPro $179.00
Administrator $399.00
Power.....$79.00

We stock a full inventory of software for Commodore, such as: Artworx, Broderbund, Commercial Data, Creative Software, Epyx, HES, MicroSpec, Nufekop, Romox, Sirius, Synapse, Thorn EMI, Tronix, UMI, Victory, Spinnaker, Rainbow & Timeworks!
CALL FOR DETAILS!

DISKETTES

MAXELL

5¼" MD-1.....$29.00
5¼" MD-2.....$39.00
8" FD-1 (SS/DD).....$39.00
8" FD-2 (DS/DD).....$49.00

VERBATIM

5¼" SS/DD $26.99
5¼" DS/DD $36.99

ELEPHANT

5¼" SS/SD $18.49
5¼" SS/DD $22.99
5¼" DS/DD $28.99

HEAD

5¼" Disk Head Cleaner.....$14.99

DISK HOLDERS

INNOVATIVE CONCEPTS

Flip-n-File 10 .....$3.99
Flip-n-File 50 .....$17.99
Flip-n-File(400/800ROM)Holder..$17.99

LJK ENTERPRISES

Atari Letter Perfect-Disk(40/80)...$79.99
Atari Letter Perfect-ROM(40 col)...$79.99
Atari Letter Perfect-ROM(80 col)...$79.99
Atari Data Perfect-ROM(80 col).....$79.99
Atari Spell Perfect-DISK $59.99
Atari Utility/MailMerge.....$21.00
Apple Letter Perfect.....$99.00
Apple Data Perfect $75.00
Apple LJK Utility $21.00
Apple Lower Case Generator ...$19.00

HANDHELD COMPUTERS

hp

41CV $209
41CX $259

HP 10C $51.99
HP 11C $69.99
HP 12C $88.99
HP 15C $88.99
HP 16C $88.99
HP 75C $749.99
HPIL Module $98.99
HPIL Cass. or Printer $359.99
Card Reader.....$143.99
Extended Function Module ...$63.99
Time Module $63.99

SHARP

PC-1500A... $165.99
PC-1250A.... $88.99

CE-125 Printer/Cassette....$128.99
CE-150 Color Printer/Cassette...$171.99
CE-155 8K RAM $93.99
CE 181 16K RAM $134.99
CE 500 ROM Library $29.99

NEC

PC-8201 Personal Computer...$599.00
PC-8221A Thermal Printer....$149.00
PC-8281A Data Recorder $99.00
PC-8201-06 8K RAM Chips...$105.00
PC-8206A 32K RAM Cartridge...$329.00

TIMEX/SINCLAIR 1000

Timex/Sinclair 1000 .....$35.99
16K Memory.....$29.99
2040 Printer $99.99
VuCalc $17.99
Mindware Printer.....$99.99

PERCOM

AT 88-S1 .....$329.00
AT 88-A2 .....$259.00
AT 88-S2 .....$529.00
AT 88-S1PD $429.00
AT 88-DDA $119.00
RFD 40-S1 .....$449.00
RFD 40-A1 .....$269.00
RFD 40-S2 .....$699.00
RFD 44-S1 .....$539.00
RFD 44-S2 .....$869.00

TEXAS INSTRUMENTS

TX 99-S1 .....$279.00

RANA

1000.....$319.00

TRAK

AT-D2.....$389.00

MEMORY BOARDS

Axlon 32K $59.00
Axlon 48K $99.00
Axlon 128K.....$299.00
Intec 32K.....$59.00
Intec 48K.....$85.00
Intec 64K.....$99.00
Intec Real Time Clock $29.00

ALIEN VOICE BOX

Atari $119.00
Apple.....$149.00

KOALA PAD

Atari.....$75.00
Apple $85.00
IBM $95.00
CBM 64.....$75.00

CONTROLLERS & JOYSTICKS

WICO

Joystick.....$21.99
3-way Joystick.....$22.99
Famous Red Ball.....$23.99
Power Grip.....$21.99
BOSS Joystick $17.99
ATARI/VIC Trak Ball.....$34.99
Apple Trak Ball.....$54.99
Apple Adapter.....$15.99
Apple Analog $37.99

KRAFT

Joystick.....$41.99
Atari Single Fire.....$12.99
Atari Switch Hitter.....$15.99
Apple Paddles.....$34.99
IBM Paddles.....$34.99
IBM Joystick $46.99

AMIGA

3100 Single $13.99
3101 Pair $19.99
Joyboard $37.99

TG

Atari Trak Ball $47.99
Apple Joystick $47.99
Apple Trak Ball.....$47.99

WEST
800·648·3311
In NV call (702)588-5654. Dept 0115
Order Status Number: 588-5654
P.O. Box 6689. Stateline, NV 89449

CANADA
800·268·4559
In Toronto call (416-828-0866. Dept 0115
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 1B
Mississauga, Ontario, Canada L5L1T1

EAST
800·233·8950
In PA call (717)327-9575. Dept 0115
Order Status Number: 327-9576
Customer Service Number: 327-1450
477 E. Third St., Williamsport, PA 17701

VISA

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.
INTERNATIONAL ORDERS: All shipments outside the continental United States must be pre-paid by certified check only. Include 3% (minimum $5.00) shipping and handling. EDUCATIONAL DISCOUNTS: Additional discounts are available to qualified Educational Institutions.
APO & FPO: Add 3% (minimum $5.00) shipping and handling.

```
840  FOR K = 1 TO HM
850  PRINT SN$(K)
860  NEXT
870  PRINT : PRINT "SELECT ONE OF THE ABOVE"

880  PRINT "(BREAK AND LIST TO EDIT THESE NU
     MBERS)."
890  GET A$
900 NB =  VAL (A$)
910  IF NB < 1 OR NB > HM THEN 750
920 P$ =  MID$ (SN$(NB),5,16)
930  GOSUB 1070
940  RETURN
950  REM ****************************
960  PRINT : PRINT "DIALING A NEW NUMBER"
970  PRINT : PRINT "TYPE IN THE PHONE NUMBER
      THAT"
980  PRINT "YOU WISH TO CALL."
990  PRINT "EXAMPLES:"
1000  PRINT "1-800-XXX-XXXX    LONG DISTANCE

1010  PRINT "112-800-XXX-XXXX  FROM LINCOLN
1020  PRINT "XXX-XXXX          LOCAL CALL
1030  PRINT
1040  INPUT "TYPE DESIRED NUMBER";P$
1050  GOSUB 1070
1060  RETURN
1070  REM ****************************
1080  REM NUMBER KNOWN AT THIS POINT
1090  REM READY TO DIAL
1100  PRINT D$"PR#2": PRINT "AT T F0 D"P$
1101  REM   TOUCH TONE, IF YOU WANT PULSE, U
     SE 'P'
1105  REM  CHANGE THE '0' TO '1' FOR FULL DU
     PLEX
1110  PRINT D$"PR#0"
1120  REM ****************************
1130  REM  YOU CAN RUN ANY TERMINAL
1140  REM PACKAGE THAT YOU WISH BY
1150  REM EDITING THE NEXT LINES.
1160  REM  PRINT D$"RUN TERMINAL PROGRAM"
1170  HOME : PRINT : PRINT "YOU NEED TO USE
     YOUR OWN"
1180  PRINT "TERMINAL PACKAGE AT THIS POINT"

1190  PRINT "(SUCH AS VIDEOTEX AND OTHERS)."

1200  GOTO 1970
1210  REM **********************
1220  REM AUTO ANSWER ROUTINE
1230  REM **********************
1240  REM
1250  HOME : PRINT : PRINT M$(2)
1260  PRINT
1270  REM MENU
1280 AN$(1) = "DO NOT ANSWER"
1290 AN$(2) = "ANSWER IMMEDIATELY"
1300 AN$(3) = "ANSWER AFTER XX RINGS"
1310  FOR SA = 1 TO 3
1320  PRINT SA;AN$(SA)
1330  NEXT SA
1340  PRINT : PRINT : PRINT "SELECT ONE OF T
     HE ABOVE"
1350  GET A$
1360  HOME : PRINT : PRINT "        "M$(2)
1370  ON  VAL (A$) GOSUB 1390,1470,1640
1380  GOTO 1210
1390  REM ****************************
1400  REM DO NOT ANSWER
1410  PRINT : PRINT "      ***"AN$(1)"***"
1420  PRINT : PRINT "THE SMART MODEM ANSWER
     FUNCTION"
1430  PRINT "IS DISABLED."
1440  PRINT D$"PR#2": PRINT "AT S0=0"
1450  PRINT D$"PR#0"
1460  GOTO 1970
1470  REM ****************************
1480  REM ANSWER IMMEDIATELY
1490  PRINT "   ***"AN$(2)"***"
1500  PRINT "THE SMART MODEM WILL ANSWER ALL
     "
1510  PRINT "CALLS IMMEDIATELY."
1520  PRINT "USE THIS TO TRANSFER CALLS THAT
     "
1530  PRINT "ARE ALREADY IN PROGRESS BETWEEN
     "
1540  PRINT "TWO INDIVIDUALS TO COMMUNICATIO
     N"
1550  PRINT "BETWEEN THEIR COMPUTERS."
1560  GOSUB 1800
1570  PRINT D$"PR#2": PRINT "AT"DP$"A"
1580  PRINT D$"PR#0"
1590  PRINT "AT THIS POINT YOU NEED TO RUN"
1600  PRINT "OR EXEC YOUR OWN RECEIVER"
1610  PRINT "PROGRAM TO INTERPRET THE"
1620  PRINT "INCOMING CALL."
1630  GOTO 1970
1640  REM **********************
1650  REM ANSWER AFTER XX RINGS
1660  PRINT : PRINT "  ***"AN$(3)"***"
1670  PRINT : PRINT "THE SMART MODEM WILL AN
     SWER ALL"
1680  PRINT "CALLS AFTER THE NUMBER OF RINGS
     "
1690  PRINT "YOU SELECT (FROM 1 TO 255 RINGS
     )"
1700  PRINT
1710  INPUT "HOW MANY RINGS (1 TO 255)";RG
1720 RG =  INT (RG)
1730  IF RG < 1 OR RG > 255 THEN 1710
1740  PRINT : PRINT "THE PHONE WILL BE ANSWE
     RED"
1750  PRINT "AFTER"RG"RINGS."
1760  GOSUB 1800
1770  PRINT D$"PR#2": PRINT "AT"DP$"S0="RG
1780  PRINT D$"PR#0"
1790  PRINT : GOTO 1590
1800  REM ****************************
1810  REM FULL OR HALF DUPLEX
1820  PRINT : PRINT "DO YOU WANT FULL OR HAL
     F DUPLEX?"
1830  PRINT "(IF YOU DO NOT KNOW, TRY HALF).
     "
1840  PRINT : PRINT "1    FULL DUPLEX"
1850  PRINT "2    HALF DUPLEX"
1860  GET A$
1870  ON  VAL (A$) GOTO 1880,1920
1880  REM FULL DUPLEX
1890 DP$ = "F1"
1900  PRINT : PRINT "FULL DUPLEX SELECTED."
1910  GOTO 1950
1920  REM HALF DUPLEX
1930 DP$ = "F0"
1940  PRINT : PRINT "HALF DUPLEX SELECTED."
1950  PRINT
1960  RETURN
1970  REM ****************************
1980  REM EXIT ROUTINE
1990  PRINT : PRINT "THIS PROGRAM HAS ENDED.
     "
2000  PRINT "THIS PROGRAM WILL GO INTO"
2010  PRINT "TERMINAL MODE IN 5 SECONDS"
2015  PRINT "-------------------------"
2020  PRINT "TYPE 'CTRL-C' FOR LOWER CASE"
2030  PRINT "TYPE 'CTRL-A' FOR SENDING LINE
     FEEDS"
2040  PRINT "TYPE 'CTRL-R' 'CTRL-X' TO EXIT
     TERMINAL MODE"
2060  PRINT "-------------------------"
2065  FOR I = 1 TO 10000: NEXT
2070  PRINT D$"PR#2": PRINT  CHR$ (9)"H"
2080  REM THIS PUTS MOUNTAIN CPS CARD
2090  REM INTO HALF DUPLEX TERMINAL MODE
2100  REM FOR FULL DUPLEX SUBSTITUTE
2110  REM  AN 'F' FOR THE 'H' IN ABOVE LINE
2120  PRINT D$"PR#0": END
```

Speedload Enhanced Apple DOS

By Enrico Colombini

Speedload requires:

Apple II or Apple II+ or Apple IIe 48K or 64K with disk drive(s) and DOS 3.3 (16 sectors).

This article describes how to build an enhanced version of Apple DOS 3.3 which performs all loading operations about four times faster then original DOS. The modified DOS is fully compatible with a 48K slave DOS, including inner routines and addresses, excepted that it cannot format new diskettes for the first time.

Why Apple DOS Is So Slow

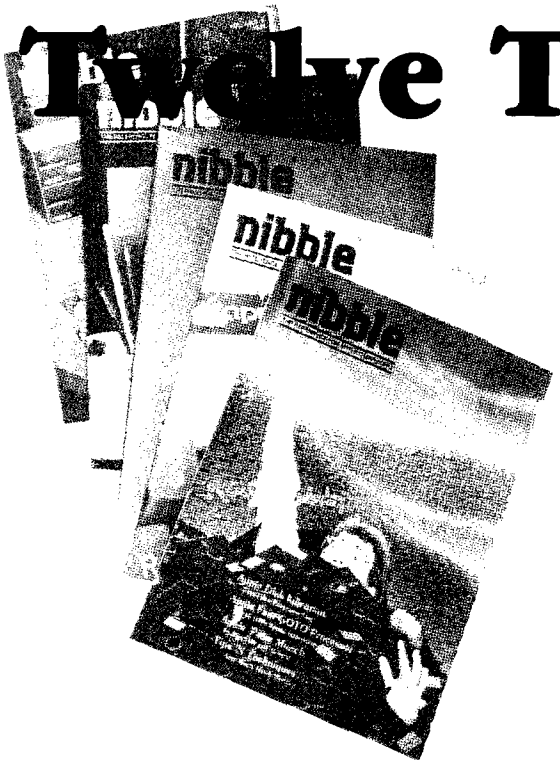
Since I started using Apple, I was puzzled by the apparently strange behaviour of the Disk Operating System. Why the DOS itself is loaded much faster than a binary file of equal length? A slave DOS 3.3 is loaded in less than 2 seconds. A file of equal size (9.25K) is loaded in about 11 seconds. Even considering the time spent in looking at index and other file informations, there is still a great difference in loading times. With an ear at the head steppin-motor and a hand on the stopwatch, I measured the average track-loading time: about 3.70 seconds. If you consider that the disk rotates at five turns per second (a turn in 200 msec), you can see that the entire track passes under the head 3.70 by 5, that is over 18 times! Why is it not possible to load the track in just one revolution of the diskette? Where is the theoretical limit? And the practical one? For the answers, we must look at the Apple hardware and at the inner structure of DOS.

Apple does not have a specialized disk controller chip as Western Digital 1791 or similar device. It has instead a little masterpiece of engineering: a "state machine" based on a PROM and a shift register. This circuit performs the hardware encoding/decoding operations to and from the R/W electronics of the disk drive. But that simplicity has a cost: the encoding/decoding process is not complete. It just sends to or receives from the disk drive a stream of 6-bit "nibbles". The remainder of the process, that is the nibbles-bytes conversion, has to be done by software.

Every track is organized as a stream of 16 sectors, 256 bytes each. In reading, the software conversion of a sector (342 nibbles to 256 bytes) takes about 10 msec. In theory, it is therefore possible to read the entire track (200 msec) and then decode it (10 msec by 16 = 160 msec) in only 360 msec. But this implies a great waste of space for the nibblized-track buffer: 342 by 16 = 5472 bytes of memory! It is not acceptable.

A more realistic approach is to read and decode one sector at a time. That is, read a sector and decode it, read the next sector and decode it, and so on. Ok, but what is the NEXT sector? While the program decodes the first

Raise your Apple's IQ Twelve Times A Year!



A One Year Subscription Brings You 12 Issues With:

Over $500 of Programs for your Home, Business, Education and Entertainment. Complete Program Listings with Instructions.

Comprehensive Articles that show what each program does, how to use it and how to type it into your Apple, Franklin ACE or other Applesoft-compatible computer.

Regular Features for the Beginner and Expert.

On The Scene
The Latest New Software/Hardware Releases.

Products! Inside and Out
Comprehensive Product Reviews.

Education Corner
Programs that help make Learning Fun.

Tips 'N Techniques
Little known programming Tricks you can Use.

Disassembly Lines
An Expert reveals the mysteries of Applesoft.

Utilities
Superchargers for Basic, DOS, Printing, and More.

Games
Arcade Fun you can Type and Run.

Note
☐ Domestic U.S. First Class subscription rate is $51.95
☐ Canada Air Mail subscripton rate is $59.95
☐ Outside the U.S. and Canada Air Mail subscription rate is $89.95
All payments must be in U.S. funds drawn on a U.S. bank.

©1983 by MicroSPARC Inc. All Rights Reserved.

Apple® is a registered trademark of Apple Computer, Inc.
ACE® is a registered trademark of Franklin Computer, Inc.

Try a NIBBLE!

Here's what some of our Readers say:

☐ *"Certainly the best magazine on the Apple!"*
☐ *"Impressed with the quality and content."*
☐ *"Programs remarkably easy to enter."*
☐ *"I'll be a subscriber for life!"*
☐ *"Your service is fantastic . . . as a matter of fact, I'm amazed!"*

Try a NIBBLE!

NIBBLE is focused completely on the Apple and Applesoft-compatible computers.

Buy NIBBLE through your local Apple Dealer or subscribe now with the Coupon or Order Card in this issue.

You'll want Back Issues Too!

Here are some examples of programs you can get:

The Investor—Stock Tracking, Reporting, and Graphing.

Recipe Box—Kitchen/Menu Management made Fun.

The Librarian—Auto Logging and Retrieval of your Disks.

Designer/Illustrator—Art/Design Creation and Composition with Graphics.

Machine Language Editor—Quick and Easy Aid for Typing and Changing M/L Programs.

And Much . . . Much More!
NIBBLE will become a permanent part of your Reference Library. Discover why 95% of NIBBLE Readers save every issue

Join more than 120,000 Apple/Ace users who say:
"NIBBLE IS TERRIFIC!"

SUBSCRIBE NOW AND SAVE $12.00 OFF THE COVER PRICE

nibble

MasterCard VISA
We accept Master Charge & Visa

Box 325, Lincoln, MA 01773 (617) 259-9710

I'll try nibble!
Enclosed is my $26.95 (for 12 issues)
(Outside U.S., see special note on this page.)

☐ **check** ☐ **money order** ☐ **bill me (U.S. only)**

Your subscription will begin with the next issue published after receipt of your check/money order.

Card # _____ Expires _____
PLEASE PRINT CLEARLY
Signature _____
Name _____
Address _____
City _____
State _____ Zip _____

Figure 1: **Sector interleaving**

Access order by R/W head of disk drive "]

0 1 2 3 4 5 6 7 8 9 A B C D E F Physical sector order

0 7 E 6 D 5 C 4 B 3 A 2 9 1 8 F Logical sector order

sector, the next physical sector (the second) passes under the head and cannot be read. Since the decoding time (10 msec) is shorter than the sector duration (12 msec approximately including header), the program can safely read the third sector. After that, it loses the fourth, reads the fifth and so on.

It is so possible to read only 8 sectors per disk revolution. Starting with the disk head at the beginning of the first sector, it takes two turns (400 msec) to read the entire track: reducing the buffer from 5472 bytes to only 342, we lost about 10% of speed. A very good compromise. Considering a head positioning and first sector finding time of 150 msec (average), we can expect to read a track in about 550 msec, or .55 seconds. The DOS track loading time is near to this value, so we can consider this time (550 msec) as the practical speed limit.

Well, Apple can load a track in .55 seconds. Yet we do not know why DOS cannot. We must find the reason in the deepest depths of DOS itself.

The Cost of Structured Programming

I like structured programming. It improves program development, testing, debugging and reading. The program occupies less memory and is much more easy to modify, even by another programmer. I make large use of it in every language, including Basic, and I am greatly rewarded: my work is much easier. All these advantages are achieved at only one cost: the program runs slower. How much slower it runs, it depends on the level of structuring. More deep the structure, more slow the program. For some time-critical programs, structuring to the deepest level can be a great disadvantage. For these programs, the programmer must find the better compromise, that is the level where it must stop structuring and start "spaghetti-programming".

Apple DOS is a deeply structured program. It is very well written and can be easily adapted to different diskette organization, that is shorter sectors, more tracks, etc. At the moment, however, the 3.3 version is widely accepted and the coming of another version is very unlikely, unless the DOS is wholly changed for new disk drives. We can therefore see if the slowness depends on the DOS structure and, if it is so, see if it is possible to reduce the structure level and so speed up the operations.

Apple DOS is made mainly of three blocks, in order of depth:

— The external level, which is the interface with Basic, monitor and user.
— The file manager, which performs all operations on diskette files.
— The RWTS, which physically reads/writes a single disk sector.

This is (roughly) the sequence of operations when Basic inputs a string from a sequential text file, already opened:

— Basic calls input routine for getting the first character. Since READ command is in effect, DOS intercepts the call.
— DOS external level calls file manager asking for a single character from active input file.
— File manager calls its internal routine "read one byte" (RD1BYT).
— RD1BYT calls a routine which, if a byte is not already available, finds and loads the next sector in the buffer (using RWTS).
— RD1BYT gets the requested byte from the buffer.
— RD1BYT calls a routine which increments the pointer (offset) in current record. In case of sequential files, records are of course one byte long. Since end of record is reached, the record counter is incremented and the offset is reset to zero.
— RD1BYT calls another routine which increments the POSITION file pointer. After that, it returns with the wanted byte to main file manager routine.
— File manager returns to external level.
— External level returns to Basic with the first character.

The above procedure is repeated for every character requested by Basic.

Just a little confused by this simplified explanation? Well, think that all increments and calculations are done in 16 bit arithmetic and you may begin to understand why sequential files reading is so slow. But, you will ask, what has that to do with loading of Basic or binary files? It has since DOS, as I said, is highly structured: it loads these files using the same above described procedure, as if they would be text files! So, Basic and binary loading is very slow too.

The SPEEDLOAD Approach

How much is the wasted time, in terms of disk sectors? When DOS is loading a Basic or binary file, after calling RWTS for reading a sector it waits for a little less than 100 msec (8 sectors or half a track) before calling the RWTS again for the next wanted sector.

I must now explain another feature of Apple DOS: sector interleaving. Using RWTS driven by a fast routine, as it happens when loading DOS, it is possible to load a sector every two, according with our former analysis. Maximum speed can be achieved only if sectors are loaded in that order. In DOS 3.3, every track is initialized with sectors in ascending order, that is starting with sector $0 and ending with sector $F (hex). But this is the physical sector order. Using a table called "interleaving table", DOS make use of the sectors as if they were in another order, the "logical order".

Figure 1 shows the correspondence between physical and logical sector numbers. The upper line represents the sector numbers in a track as they are physically written on the diskette during initialization process. The lower line shows the sector numbers as they are used by the DOS (the RWTS). DOS stores sectors in descending logical order, that is starting with sector $F and proceeding towards sector $0. Every sector is stored in the physical

sector located through the interleaving table. For example, logical sector $F is stored in physical sector $F, logical $E in physical $2, $D in $4 and so on, as stated by the table in fig. 1. As DOS always uses sectors in logical order, you can forget the upper line and look only at the lower line (logical order).

When loading DOS itself, after reading sector $7 it must be decoded. During this operation, sector $E passes under the head, unnoticed. When RWTS is ready for reading sector $6, it is just this sector which passes under the head, with very little delay. The loading time is thus optimized. The situation is very different when loading a Basic file (it is the same matter with a binary file): the time needed by the file manager for processing sector $7 is so great that in the meanwhile sectors $E,$6,$D,$5,$C,$4,$B and $3 pass under the head. When at last DOS is ready for reading sector $6, that one is already passed by and it has to let sectors $A,$2,$9,$1,$8,$F,$0,$7 and $E pass under the R/W head before it can at last read sector $6. 17 sectors to wait: a very long time!

In the book *Bag of tricks* by Don Worth and Peter Lechner (published by Quality Software) you can find a similar description of the problem. The authors call the time wasted in waiting the next sector to read "rotational delay" and the time wasted in processing a sector "file manager overhead". Searching for a solution, they discarded (rightly) the idea of modifying the interleaving table, because DOS and diskettes thus created were incompatible with standard ones. They suggest to modify the physical sectors order, in such a way that 8 sectors lay between two logical consecutive sectors. It is a good idea, but it has some disadvantages. First, it requires to re-init the diskette. Second, the majority of copy programs, and the more reliable ones, will destroy the work and restore the standard sector order on the copy. Third, all that work can achieve only a 45% gain in loading speed, in the best case. If we might read a sector every two, as in the DOS loading, we can achieve a much greater improvement. But this implies rewriting part of file manager. Well, if there is no other way, we must follow this one.

There is a problem: after loading and decoding a sector, there is no time for copying it in memory at the right place. It were therefore necessary to read it directly in memory. But this is not ever possible, as some sectors are only partially written and unwanted bytes must not be placed in memory. Looking closer at the file structure, it can be seen that only the first and the last sector does not contain only bytes to be copied in memory. The first because it begins with address/length informations and the last if it is not exactly filled (which is unlikely to happen). All the sectors in between can be loaded directly in memory without any problems.

That is the theory of SPEEDLOAD: loading the first sector in the normal way, loading all the subsequent ones directly in memory, excepted the last, if it is not wholly filled. Obviously, there is no gain in loading a very short file and in preliminary operations (index looking, etc.), but there is a great speed improvment as the file size grows up, as you can view from figure 2. Since SPEEDLOAD trick is in the DOS, it do not requires any modifications on diskette formatting and can read any standard diskette. The modified DOS can be put on the diskette and it will be booted exactly as the standard DOS. Let us go to the practical implementation of SPEEDLOAD.

***Figure 2:* Approximate binary file loading times (in seconds)**

File size	Normal DOS	Speedload DOS
1K (1024 bytes)	3.1	2.5
2K (2048 bytes)	4.2	2.7
4K (4096 bytes)	6.5	3.7
8K (8192 bytes)	9.8	3.9
16K (16384 bytes)	17.6	5.7
32K (32768 bytes)	32.3	8.3

Inside The File Manager

Since SPEEDLOAD must deal with Basic and binary files, we need to look closely at the loading process. The involved commands are LOAD, BLOAD, RUN and BRUN. First, notice that RUN and BRUN make use respectively of LOAD and BLOAD routines, so we can look only at the latter ones. Let's look at the DOS loading process:

- External level calls file manager for reading file informations: loading address and length for binary files, length only for Basic files.
- External level calls file manager for reading a block of bytes of specified length.
- External level sets required parameters and, if needed, starts the loaded program.

In all loading operations, file manager make use of the formerly outlined RD1BYT routine. But for loading a block RD1BYT is called through a "read L bytes" (RDLBYT) routine which sets L (the required length) and jumps in a general purpose routine (read/write a range of bytes). It is therefore possible to alter this jump and redirect it to a hi-speed block loading routine.

Since the 48K slave version of DOS 3.3 is universally accepted, I decided to make SPEEDLOAD not relocatable, thus keeping it smaller.

After a careful analysis (using some DOS sources and the Quality Software's *Beneath Apple DOS*), I determined the input parameters and conditions of RDLBYTE. These are outlined in figure 3. The new routine must respect these conditions in order to maintain compatibility with standard DOS.

The SPEEDLOAD routine

The SPEEDLOAD routine is very little structured, because speed is needed. There are only 2 msec (or little more) available between RWTS calls.

Figure 4 is the listing of SPEEDLOAD routine, (in LISA 2.5 format). Notice the assembly address: the SPEEDLOAD routine take the place of the DOS 3.3 FORMAT routine, that is the routine which physically initializes a new diskette, writing the address marks on it (and empty data too). The FORMAT routine is therefore no more available, but this do not means that it is impossible to INIT a diskette. Since the INIT routine is not modified, it is always possible to re-INIT a previously

Figure 3: **RDLBYT input variables (for 48K slave DOS). All 2'byte variables are in the L'H order.**

Memory locations:

Loc.	Name	Contents and/or meaning
$B5BB	FMOPC	$03 = READ (FMNGR opcode)
$B5BC	FMSUBC	$02 = R/W RANGE (FMNGR subcode)
$B5BD	FMPAR+0	Record index. In practice, position in
	FMPAR+1	file ¢ 1; it is $01 if Basic file, $03 if binary.
$B5BF	FMPAR+2	Byte offset in the record; always 0.
	FMPAR+3	
$B5C1	FMPAR+4	No. of bytes to read (FILEN)
	FMPAR+5	
$B5C3	FMPAR+6	Before jumping to SPEEDLOAD
	FMPAR+7	routine, it will contain the loading address (now in Y,X)
$B5C5	FMPAR+8	FMNGR return code
$B5C6	FMPAR+9	Not used
$B5C7	FMPAR+10	Address of FMNGR work area buffer
	+11	
$B5C9	FMPAR+12	Address of T/S list buffer, which
	+13	contains the first T/S list sector
$B5CB	FMPAR+14	Address of sector buffer, which
	+15	contains the first data sector

CPU registers:

A: don't care
Y,X: loading address (H,L) to be stored in FMPAR+6,+7

Zero page variables available:

$42,$43 only.

used diskette (if it is not damaged). It is not possible to INIT a new diskette for the first time, because the routine which writes the address marks is no longer within DOS. A direct call to RWTS Format returns with a "format error" indication.

A diskette must be INITed for the first time using a normal 3.3 DOS. After that, it can be re-INITed with SPEEDLOAD DOS, unless an unwanted exposition to a magnetic field has destroyed the formatting. More information about INIT in the next paragraph (installation).

The routine is 257 ($101) bytes long. A more detailed analysis of DOS might have allowed to further reducing the routine size, making use of some DOS subroutines, but I preferred a self-contained routine. The only external call is for RWTS. You might be surprised to see a single zero page pointer (PTR in the listing) used alternately for two purposes, but I chose to use only locations used by the standard file manager routine, thus insuring total compatibility in any case.

The DCM directive instructs the assembler to save the produced object code under the name SPL.M.

Installing SPEEDLOAD

Once you have assembled the SPEEDLOAD routine, you must build a SPEEDLOAD DOS, following this procedure:

— Boot a standard 3.3 DOS.
— Put in a new (blank) diskette.
— INIT the blank diskette.
— Insert the diskette with SPL.M
— BLOAD SPL.M, A$BEAF
— Remove the diskette with SPL.M
— CALL-151 (go to monitor)
— A477: 4C BA BE (link SPL routine)
— AE93: EA EA EA (unlink Format routine)
— CTRL-C (return to Basic)
— Put in again the blank diskette.
— INIT the blank diskette again.

You now have a diskette with SPEEDLOAD DOS on it. Transferring SPEEDLOAD DOS to another diskette can be achieved in two ways:

1.) INIT a new diskette with standard DOS, then boot SPEEDLOAD DOS and re-INIT the diskette.
2.) Transfer tracks 0-2 from SPEEDLOAD diskette to normal diskette, using some copy program.

The second way allows you to install SPEEDLOAD without erasing the data on the diskette, if you have a track copy program.

If you want to copy a SPEEDLOAD diskette, you can do it with COPYA, but the destination diskette must have been previously INITed with normal DOS, or you will get an I/O ERROR message.

Conclusions

Although there are many speeded-up DOS available on the market, SPEEDLOAD DOS is the only one, for that I know, which is fully compatible with normal DOS in inner addresses, POKEs, variable usage and so on. SPEEDLOAD works only in loading process, and that is its major limitation, but that is a constraint if you want to keep compatibility with DOS 3.3 (and, beside that, I do not wanted to rewrite all the DOS). However, if you have a big program, or you made use of chaining or machine-language routines, SPEEDLOAD can save you lots of time, in program development and usage.

I tested SPEEDLOAD with over 200 programs and I do not found any incompatibility, even with programs which make extensive use of the file manager, as FID or Microsoft's TASC compiler. I confess that I removed the protection system from many of my game diskettes, only for installing SPEEDLOAD and avoiding frustrating delays. I hope we do not have to see again the hateful "please wait while loading...", now that SPEEDLOAD is available. MICRO

Biography

After working five years in designing specialized circuits for FM broadcasting, I founded a little firm which is now growing in the European industrial micro-board market. Since January 1983 I work as free-lance programmer and writer (mainly for Jackson publishing group, based in Milan). My first interest in programming are high level multi-player games and didactical applications. I am responsible for small computers in ABRS (a research association) and ASTROFISMA (a didactical one for young people). I work with an Apple II, two drives and silentype.

```
                        ;
                        ;
                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXX
                        ;X                        X
                        ;X      SPEEDLOAD         X
                        ;X                        X
                        ;XXXXXXXXXXXXXXXXXXXXXXXXX
                        ;
                        ;
                        ; DOS 3.3 SPEED ENHANCER
                        ;
                        ;
                        ; BY E.Colombini
                        ; Brescia-Italy
                        ;
                        ;
                        ; Copyright (C)
                        ; MICRO INK 1983
                        ;
                        ;
                        ;
                        ; * DOS 3.3 ROUTINES *
                        ;
B7B5-                   RWTS    EQU $B7B5       ;R/W Track/sector
A2EA-                   CLOSE   EQU $A2EA       ;Closes file(s)
B65E-                   DOSERR  EQU $B65E       ;Error (code in FMRETC)
BEAF-                   FORM    EQU $BEAF       ;RWTS Format entry
                        ;
                        ;
                        ; * DOS 3.3 VARIABLES *
                        ;
                        ; (valid on RDLBYT+6)
                        ;
B5BD-                   RECORD  EQU $B5BD       ;Record number in file (L,H)
B5C1-                   FILEN   EQU $B5C1       ;Length of range to read
B5C3-                   MEMPT   EQU $B5C3       ;Memory pointer
B5C5-                   FMRETC  EQU $B5C5       ;File manager return code
B5C9-                   TSBUF   EQU $B5C9       ;T/S list buffer address
B5CB-                   SECBUF  EQU $B5CB       ;Sector buffer address
B7E8-                   IOB     EQU $B7E8       ;RWTS I/O control block
B5F9-                   VOLUME  EQU $B5F9       ;Disk volume
                        ;
                        ;
                        ; * SPEEDLOAD VARIABLES *
                        ;
                        ; (Borrowed from DOS vars)
                        ;
B5BF-                   TSPT    EQU $B5BF       ;pointer in TSBUF
B5E0-                   TY      EQU $B5E0       ;Temp. storage for Y reg.
0042-                   PTR     EPZ $42         ;Temp. pointer
                        ;
                        ;
                        ;
                                ORG FORM        ;Over RWTS Format routine
                                OBJ $800        ;For Lisa assembler
                        ;
                        ;
                        ;
                        ; * "FORMAT ERROR" ROUTINE *
                        ;
                        ; This routine takes the place of
                        ; the normal Format routine of RWTS.
                        ; SPEEDLOAD cannot INIT diskettes
                        ; for the first time.
                        ; Any attempt of INIT a new diskette
                        ; causes an error (I/O error). Any
                        ; attempt to use the Format routine
                        ; of RWTS falls in this routine, with
                        ; the same result.
                        ;
BEAF- A9 08     FORMER  LDA #$08        ;Code 08: Format error
BEB1- A0 0D             LDY #$0D        ;Put code in IOB
BEB3- 91 30             STA (48),Y
BEB5- BD 88 C0          LDA $C088,X     ;Turn off motor
BEB8- 38                SEC             ;Flag error
BEB9- 60                RTS             ;And return.
BEBA-                   ;
BEBA-                   ;
BEBA-                   ;       *** MAIN ROUTINE ***
BEBA-                   ;
BEBA-                   ; Reads a range of bytes. Must be linked
BEBA-                   ; to main DOS (file manager) program at
BEBA-                   ; RDLBYT+6 ($A477).
BEBA-                   ; Works ONLY with a slave DOS 3.3 48K.
BEBA-                   ;
BEBA-                   ;IN:  RECORD=Last used byte in file
BEBA-                   ;     FILEN=No. of bytes to read
BEBA-                   ;     MEMPT=RAM destination address
BEBA-                   ;     TSBUF=Address of T/S buffer
BEBA-                   ;     SECBUF=Address of sector buffer
BEBA-                   ;     IOB=OK excepted T,S,V,C
BEBA-                   ;OUT: JMP CLOSE if all OK
BEBA-                   ;     JMP DOSERR if loading error
BEBA-                   ;
BEBA- A9 0E             SPL     LDA #$E         ;Start T/S list scanning from
BEBC- 8D BF B5          SPL0    STA TSPT        ;second sector (the next)
BEBF- A9 00             SPL1    LDA #0          ;By now, no errors
BEC1- 8D C5 B5                  STA FMRETC
BEC4- AC BD B5                  LDY RECORD      ;Y=1st useful byte in SECBUF
BEC7- C8                        INY             ;(with correction)
BEC8- AD CB B5          SPL2    LDA SECBUF      ;SECBUF in PTR
BECB- 85 42                     STA PTR
BECD- AD CC B5                  LDA SECBUF+1
BED0- 85 43                     STA PTR+1
BED2- B1 42             SPL3    LDA (PTR),Y     ;Read byte from SECBUF
BED4- AE C3 B5                  LDX MEMPT       ;MEMPT in PTR
BED7- 86 42                     STX PTR         ;(uses only one pointer)
BED9- AE C4 B5                  LDX MEMPT+1     ;(for total compatibility)
BEDC- 86 43                     STX PTR+1
```

```
BEDE 8C E0 B5   111          STY TY             ;Save Y
BEE1 A0 00      112          LDY #0
BEE3 91 42      113          STA (PTR),Y        ;Put byte in memory
BEE5 AC E0 B5   114          LDY TY             ;Restore Y
BEE8 EE C3 B5   115          INC MEMPT          ;MEMPT=MEMPT+1
BEEB D0 03      116          BNE SPL2
BEED EE C4 B5   117          INC MEMPT+1
BEF0 AD C1 B5   118 SPL2     LDA FILEN          ;FILEN=FILEN-1
BEF3 D0 03      119          BNE SPL3
BEF5 CE C2 B5   120          DEC FILEN+1
BEF8 CE C1 B5   121 SPL3     DEC FILEN
BEFB AD C1 B5   122          LDA FILEN          ;If FILEN=0, done
BEFE 0D C2 B5   123          ORA FILEN+1
BF01 F0 2C      124          BEQ SPLR           ;and exit.
BF03 C8         125          INY                ;If not, ready for next byte
BF04 D0 C2      126          BNE SPL1           ;If not end of sector, again
BF06 AD C2 B5   127 SPL4     LDA FILEN+1        ;Full sectors available
BF09 D0 0D      128          BNE SPL5           ;Yes, load them the fast way
BF0B AD CB B5   129          LDA SECBUF         ;No, load last in SECBUF
BF0E AE CC B5   130          LDX SECBUF+1
BF11 20 32 BF   131          JSR LNEXT          ;Load last sector
BF14 A0 00      132          LDY #0             ;Y on SECBUF start
BF16 F0 B0      133          BEQ SPL1           ;move it in RAM (JMP)
BF18 AD C3 B5   134 SPL5     LDA MEMPT          ;Load sec. directly in RAM
BF1B AE C4 B5   135          LDX MEMPT+1
BF1E 20 32 BF   136          JSR LNEXT          ;Load full sector
BF21 EE C4 B5   137          INC MEMPT+1        ;Next memory page
BF24 CE C2 B5   138          DEC FILEN+1        ;256 bytes loaded
BF27 AD C1 B5   139          LDA FILEN          ;Done
BF2A 0D C2 B5   140          ORA FILEN+1
BF2D D0 D7      141          BNE SPL4           ;If not, another loop
BF2F 4C EA A2   142 SPLR     JMP CLOSE          ;That's all, folks.
BF32            143 ;
BF32            144 ;
BF32            145 ;
BF32            146 ; # SUBROUTINE LNEXT #
BF32            147 ;
BF32            148 ; Loads next sector directly in memory.
BF32            149 ; If end of current T/S list sector is
BF32            150 ; reached, then loads next list sector.
BF32            151 ;
BF32            152 ;IN:  A,X=Loading address (L,H)
BF32            153 ;     IOB, TSBUF=As in main routine
BF32            154 ;     TSPT=Current pos. in TSBUF
BF32            155 ;OUT: RTS if all OK
BF32            156 ;     JMP DOSERR if load error
BF32            157 ;     or invalid sector (T=0)
BF32            158 ;
BF32 8D F0 B7   159 LNEXT    STA IOB+8          ;Loading address in IOB
BF35 8E F1 B7   160          STX IOB+9
BF38 AC BF B5   161          LDY TSPT           ;Y=Index in TSBUF
BF3B AD C9 B5   162          LDA TSBUF          ;TSBUF in PTR
BF3E 85 42      163          STA PTR
BF40 AD CA B5   164          LDA TSBUF+1
BF43 85 43      165          STA PTR+1
BF45 B1 42      166          LDA (PTR),Y        ;Track in IOB
BF47 8D EC B7   167          STA IOB+4
BF4A C8         168          INY
BF4B B1 42      169          LDA (PTR),Y        ;Sector in IOB
BF4D 8D ED B7   170          STA IOB+5
BF50 20 87 BF   171          JSR READ           ;Read sector
BF53 EE BF B5   172          INC TSPT           ;TSPT=TSPT+2
BF56 EE BF B5   173          INC TSPT
BF59 D0 2B      174          BNE LNXR           ;If within TSBUF, end
BF5B AD C9 B5   175          LDA TSBUF          ;TSBUF address in PTR
BF5E 85 42      176          STA PTR
BF60 AD CA B5   177          LDA TSBUF+1
BF63 85 43      178          STA PTR+1
BF65 A0 01      179          LDY #1             ;point to T/S link
BF67 B1 42      180          LDA (PTR),Y        ;Track in IOB
BF69 8D EC B7   181          STA IOB+4
BF6C C8         182          INY
BF6D B1 42      183          LDA (PTR),Y        ;Sector in IOB
BF6F 8D ED B7   184          STA IOB+5
BF72 AD C9 B5   185          LDA TSBUF          ;Load it in TSBUF
BF75 8D F0 B7   186          STA IOB+8
BF78 AD CA B5   187          LDA TSBUF+1
BF7B 8D F1 B7   188          STA IOB+9
BF7E 20 87 BF   189          JSR READ           ;Read next T/S list sector
BF81 A9 0C      190          LDA #$C            ;Point TSPT on first file
BF83 8D BF B5   191          STA TSPT
BF86 60         192 LNXR     RTS                ;And return.
BF87            193 ;
BF87            194 ;
BF87            195 ;
BF87            196 ; * SUBROUTINE READ *
BF87            197 ;
BF87            198 ; Reads a sector using RWTS
BF87            199 ;
BF87            200 ;IN:  IOB OK excepted C,V
BF87            201 ;OUT: RTS if OK (and FMRETC=0)
BF87            202 ;     JMP DOSERR if loading error
BF87            203 ;     or invalid sector (T=0)
BF87            204 ;
BF87 AD EC B7   205 READ     LDA IOB+4          ;If track=0, error!
BF8A F0 1B      206          BEQ READER
BF8C AD F9 B5   207          LDA VOLUME         ;VOLUME in IOB
BF8F 49 FF      208          EOR #$FF
BF91 8D EB B7   209          STA IOB+3
BF94 A9 01      210          LDA #$01           ;Command=Read
BF96 8D F4 B7   211          STA IOB+$C
BF99 A0 E8      212          LDY #IOB           :IOB address in Y,A
BF9B A9 B7      213          LDA /IOB
BF9D 20 B5 B7   214          JSR RWTS           ;Read, at last!
BFA0 A9 FF      215          LDA #$FF           ;Reset VOLUME
BFA2 8D F9 B5   216          STA VOLUME
BFA5 90 08      217          BCC REAR           ;if no errors, end
BFA7 A9 08      218 READER   LDA #$08           ;if error, set
BFA9 8D C5 B5   219          STA FMRETC         ;I/O ERROR code
BFAC 4C 5E B6   220          JMP DOSERR         ;And leave (bye).
BFAF 60         221 REAR     RTS                ;End of read.
BFB0            222 ;
BFB0            223 ;
BFB0            224 ; End of SPEEDLOAD routine source code.
BFB0            225 ;
BFB0            226 ;
BFB0            227          DCM "BSAVE SPL.M,A$800,L$101"
BFB0            228 ;
BFB0            229 ;
BFB0            230 END      END
```

MICRO™

FOR YOUR APPLE II

Industry standard products at super saver discount prices

SOFTWARE

	List	SGC
ARTSCI		
Magicalc	$149.00	$ 99.00
Magicwindow	99.00	65.95
Magicword	99.00	60.95
Special, All 3 of above	347.00	219.95
BRODERBUND		
Payroll	$395.00	$255.95
Acct/Rec.	395.00	255.95
Arcade Machine	59.95	39.95
Choplifter	34.95	24.95
Sea fox	29.95	20.95
Galaxy Wars	24.95	20.95
DATAMOST		
Property Mgr.	$295.00	$194.95
Real Estate Investment Package	179.95	118.95
Tax Beater	129.95	89.95
Zaxxon	39.95	27.95
Casino	39.95	27.95
Conquering Worlds	29.95	20.95
Mating Zone	29.95	20.95
Pandoras Box	29.95	20.95
Series Baseball	29.95	20.95
INFOROM		
Zork 1, 2, 3	$ 39.95	$ 27.95
Planet Fall	49.95	32.95
Starcross	39.95	27.95
PEACHTREE		
Peachcalc	$150.00	$ 92.95
PeachPak (G1/Ar/Ap/)	395.00	226.95
Mailing List Mgr.	250.00	151.95
SENSIBLE SOFTWARE		
Medical dictionary	$ 99.95	$ 72.95
DOS Plus	25.00	18.95
Legal Dictionary	99.95	72.95
SIR TECH SOFTWARE		
Legacy of Llylgamyn	$ 39.95	$ 29.95
Wizardry	49.96	34.95
Galactic Attack	29.95	21.95
SIRIUS		
Bandits	$ 34.95	$ 24.95
Epoch	34.95	24.95
Fly Wars	29.95	20.95
Beer Run	29.95	20.95
Outpost	29.95	20.95
Snake Byte	29.95	20.95
Sneakers	29.95	20.95
Joy Port	49.95	37.95
STRATIGIC SIMULATIONS		
Germany	$ 59.95	$ 42.95
North Atlantic 1986	59.95	42.95
Rapid Deployment Force	34.95	25.95
Road to Gettysburg	59.95	42.95
Computer Baseball	39.95	32.95
Computer Quarterback	39.95	32.95
Bomb Alley	59.95	42.95
Napoleons Campaigns	59.95	42.95

SPECIAL AND NEW

FRANKLIN COMPUTER

With 48K RAM, 1 Disk Drive, Disk Controller Card, 12" Amber Monitor, Additional 64K RAM, Magicalc, Magicword, Magicwindow, Magicspeller

Special Price **$1949.95**

"New" from Prometheus Products

ProModem® 1200 Bell 212A 300/1200 Baud Modem

A. Can be expanded into a telecommunication system as a stand-alone package.
B. Real time clock and calendar
C. Help Commands
D. Built-in diagnostics
E. Auto dialer
F. Programmable Intelligent Dialing
G. RS 232 interface
H. Internal Power Supply

OPTIONS

On line phone directory
Expandable buffer from 16K to 64K

SPECIAL ..... **$ 475.00**

BASIS 108

128K computer ..... **SPECIAL $915.00**

Also available with CPM 3.0 or DOS dual drives monitor, perfect series software or Magicalc, Magicwindow and Magicword. Call.

PROMETHEUS PRODUCTS

16K Card	**$ 60.00**
64K Expand-A-Ram	**$ 225.00**
128K Expand-A-Ram	**$ 275.00**
Graphitti Card (Specify Printer)	**$ 89.00**
P.S. 16K Buffer Card Expandable to 64K	**$ 105.00**
Applesurance Card	**$ 99.00**
VersaCard Four cards in one 1. Serial 2. Parallel 3. Real Time clock and calendar 4. BSR Port	**$ 159.95**
With Graphics	**$ 199.95**
PRT-1 Printer Card	**$ 79.00**

PRINTERS

NEC 8023	**$ 395.00**
with interface card and cables	**$ 449.00**
OKI 82A 80 col./120cps	**$ 439.00**
OKI 83A 136 col./120 cps	**$ 695.00**
Brother Letter Quality	**$ 889.00**
Transtar 315 Color Printer	**$ 469.95**

ACCESSORIES

T.G. Joystick	**$ 42.95**
Select a Port	**$ 39.95**
System Saver Fan	**$ 69.95**
Super Fan 11	**$49.95**
Slimline Disk Drives	**$ 225.00**
Disk Drives Full Ht.	**$ 195.00**
Floppy Disc Box 10	**$ 25.00**
Floppy Disc w/order	**$ 19.90**

BUSINESS SPECIAL

Includes 80 Column Z-80 Card • 16K Card • Wordstar • Supercalc

Special **$795.00**

D.O.S. Programming SPECIAL

80 column Card • Magicalc • Magicwindow • Magicspeller • 16K Card **Special** **$449.00**

3-CARD SYSTEM

80 col/16K card/Z-80 card **Special** **$319.95**

MODEMS

Smartmodem 300	**$219.95**
Smartmodem 1200	**$499.00**
D-Cat Modem	**$164.00**

MONITORS

14" Composite Color High Res	**$260.00**
Amber 12"	**$129.95**
Taxan RGB color	**$395.00**
USI Hi/Res green	**$141.95**

80 COLUMN CARDS

Smarterm 11	**$149.95**
Videx Video Term	**$215.00**
Ultra Term	**$289.00**
View Master	**$139.00**
Apple IIe 64K, 80 col.	**$ 99.00**

Z-80 CARDS

Microsoft Softcard	**$229.95**
Z-80 Plus	**$125.00**
Z-Card 11	**$131.95**

EDUCATIONAL SOFTWARE

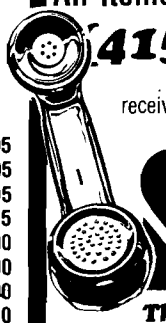
Spelling/Reading Primer	**$ 29.95**
Perception	**$ 20.95**
Counting Bee	**$ 23.95**
Word Scrambler	**$ 15.95**
Fractions	**$ 36.95**
Intro. Algebra	**$ 21.95**

BUSINESS SOFTWARE

Wordstar	**$244.95**
Spellstar	**$159.95**
WStar/MMerge	**$395.00**
WStar/SStar	**$422.00**
Supercal	**$129.95**
PSF File	**$ 88.75**
PSF Graph	**$ 88.75**
PSF Report	**$ 88.75**
PSF 3PAK	**$246.75**

All equipment shipped factory fresh. Manufacturers' warranties included. California customers add 6½% tax. Include payment by personal check, money order, or cashier's check with order and SGC will pay shipping charge. Call for amount of shipping charge when paying by credit card.

All items are normally in stock



(415) 490-3420

... And we'll be here to help after you receive your order. Feel free to call the SGC Technical Staff for assistance.

SGC

The mail order specialists

342 Quartz Circle, Livermore, CA 94550

Product Name: **Printmate 150G Printer**
Equip. req'd: Serial, Parallel or IEEE interface
Price: ?
Manufacturer: Micro Periphals, Inc.
4426 Century Drive
Salt Lake City, UT 84107

Description: A high quality dot-matrix printer with bi-directional printing, true descenders and one-pass underlining capability, the 150G prints 100 CPS, 132 characters per line and has a 1K (expandable to 8K) buffer. While printing normally with a 5 × 9 dot matrix, a serif font is included for letter quality type in a 11 × 9 matrix. Dot addressable graphics are supported, and an Ap-pak is available for Apple owners with full software support for screen dumps including different sizes, alignment and printing two pictures at once. Special fonts may be down loaded into RAM or stored in ROM. An optional external keypad is available for setting printer options and is backed up with a battery for remembering the settings.

Pluses: A very fast, very high quality printer which makes the imported models pale in comparison. If you want a better quality printer at a comensurate price, this is the printer I recommend. I have used it for all my printing for 4 months and have not had a single problem.

Minuses: None.

Documentation: An 80+ page technical manual and a 50+ page Ap-pak reference manual are included. They are terse, well written manuals. If you like chattiness, buy an import.

Skill level required: No prior knowledge necessary.

Reviewer: Phil Daley

Product Name: **Money Tool**
Equip. req'd: Apple II+ or IIe, disk drive, printer
Price: $59.95
Manufacturer: Howard W. Sams & Co., Inc.
4300 W. 62nd St.
Indianapolis, IN 46268

Description: This product is an excellent checkbook/budgeting program. It is set up to automatically categorize expenses and apply them against deposits. Additionally, features are provided to accommodate payroll deduction storage for end-of-year tax accounting.

Pluses: The program is incredibly fast for what it has to do. Categorized summations are available for any desired time period. Hard copy is neatly structured and easily interpreted.

Minuses: The program is overpriced. I've seen similar home finance utilities for half as much. Functionally, the only feature I found offensive was the inability to display on the screen all the reports that could be printed.

Documentation: The documentation is entirely adequate. The tutorial is written clearly and is easily followed. Doing so, one can learn everything in about two sittings.

Skill level required: A new Apple owner would have no problem using the program.

Reviewer: Chris Williams

Product Name: **Apple Record Manager**
Equip. req'd: Apple II or II+ with one disk drive and 48K
Price: $40.00
Manufacturer: Connecticut Info Systems
218 Huntington Road
Bridgeport, CT 06608

Description: Creates and saves text files, which you can then edit, append, rename, or delete. This Menu-driven program allows you to set up sequential data files of variable length. You name the files and fields and can quickly view/edit data prior to printing or saving.

Pluses: One-keystroke commands make operation smooth and simple. A quantity of utilities perform all needed functions from the menu.

Minuses: Error-handling is only marginal and might trip up beginners. The screen often becomes cluttered as Record Manager is a stripped-down package and skimps on niceties, such as screen formatting.

Documentation: No written instructions are provided, but upon booting the disk, on-screen instructions are displayed, and they are satisfactory for acquainting the user with information needed to operate the program.

Skill level required: Some knowledge of text files is useful but not mandatory. After some stumbling around, a beginner will get the hang of things.

Reviewer: Mike Cherry

Product Name: **Micro Illustrator**
Equip. req'd: Apple II w/48K and Applesoft
Recommended: Koala Touch Tablet
Price: $129.95 with Tablet

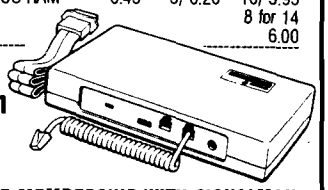
(continued)

Hewlett Packard

Write or call for prices.

SPECIALS on INTREGATED CIRCUITS

6502	7.45	10/ 6.95	50/ 6.55	100/ 6.15
6520 PIA	5.15	10/ 4.90	50/ 4.45	100/ 4.15
6522 VIA	6.45	10/ 6.10	50/ 5.75	100/ 5.45
2716 EPROM		4.45	5/ 4.20	10/ 3.95
2532 EPROM		5.90	5/ 5.75	10/ 5.45
6116 2KX8 CMOS RAM		6.45	5/ 6.20	10/ 5.95
4116 RAM				8 for 14
4164 RAM				6.00



Anchor Automation Signalman Modems

FREE SOUBCE MEMBERSHIP WITH SIGNALMAN

All Signalman Modems are Direct Connect, and provide the best price-performance values.

Dealer and OEM inquiries invited

Volksmodem with computer cable	**(99)**	**68**
Mark VI for IBM Personal Computer	(279)	139
Mark VII Auto Dial/Auto Answer	(159)	99
Mark XII Smart Model 1200/300	(399)	299
OC HAYES Smartmodem		**219**
DC Hayes Smartmodem 1200/300		529

PROM QUEEN for VIC	**170**
Apple Emulator for Commodore 64	**Call**
STAT Statistics Package for C64	**95**
Solid Oak 2 Level Stand for C64 or VIC	**29**
C64/VIC Switch (networking)	129
BACKUP V1.0 tape copier for C64 or VIC	**20**
CARDBOARD/6 Motherboard - VIC	64
CARDBOARD/5 Motherboard - C64	56
CARD PRINT G Printer Int. with Graphics	72
CARD PRINT B Printer Interface—C64/VIC	40
CARDBOARD/3s Motherboard - VIC	22
CARDCO C64/VIC Calculator Keypad	32
CARDRAM/16 RAM Expansion - VIC	44
Complete CARDCO Line in stock	
CIE and VIE IEEE Interfaces in stock	
MSD SuperDrive for C64 or IEEE	**365**
MAE Assembler for C64	50
Koala Touch Tablet—C64 or VIC	**79**
CBC 4/12 12-bit A to D Board + Software	**199**
Multiplan for C64	**79**
Dust Cover for C64 of VIC	6

APPLE—FRANKLIN ITEMS

FRANKLIN—complete line in stock	
QUENTIN Drives for Apple/Franklin	**219**
Swapper Stopper	**26**
automatic switch between paddles and joystick	
KRAFT Apple Joystick	**40**
Kraft Apple Paddle Pair	30
Koala Touch Tablet-Apple/Franklin	**90**
SPINNAKER Software in stock	
Broderbund Software in stock	
16K RAM Card for Apple	59
Multiplan—Microsoft	**185**
Solid Oak 2 Level Stand for Apple	**29**
Serial Card for Apple	89
MCP RAM/80 column card for IIe (AP/TXT)	**139**
Z80 Softcard and CP/M (Microsoft)	235
RANA Elite I with Controller	389
Parallel Printer Interface/Cable	69
Microtek and MCP Interfaces in stock	
Grappler + Interface	129
Kraft Products for Apple in stock	
DC Hayes Micromodem II	299
PFS: File	**95**
PFS: Report	**95**
Videx 80 Column Card	209
Apple Blue Book	**19**

commodore

See us for Personal, Business, and Educational requirements. Educational Discounts available.

PETSCAN I $245 base price

Allows you to connect up to 30 CBM/PET Computers to shared disk drives and printers. Completely transparent to the user. Perfect for schools or multiple word processing configurations. Base configuration supports 2 computers. Additional computer hookups $100 each.

COMPACK/STCP $115

Intelligent Terminal Package for PET, CBM, C64

Includes ACIA Hardware / STCP Software

SCREENMAKER 80 Column Adapter for C64 145

Provides big screen capability for business applications.

GENESIS Computer Corp

VICcontroller (for C64 as well) 50

combine with BSR modules for home or business control

COMSENSE Remote Sensing Adapter for C64 or VIC 35

COM VOICE Synthesizer for C64 or VIC 139

includes software for text to speech, pitch, etc.

COM CLOCK Real Time Clock with battery backup 45

VIC 20 Products and Software in stock

Thorn EMI Software		**UMI Software**	
ABACUS Software		**HES Software**	
16K RAM for VIC	64	Vanilla Pilot	27

VICTORY Software for VIC and C64

Metamorphosis	16	Creator's Revenge	16
Labyrinth of Creator	16	Galactic Conquest	16
Kongo Kong	16	Annihilator	16
Chomper Man	16	Grave Robbers	13
Bounty Hunter	16	Adventure Pack I or II	16

PAPER CLIP Ward Precessor	**99**
ORACLE Data Base from Batteries Included	**109**
Super BusCard from Batteries Included	**159**
Commodore 64 Programmers Reference Guide	16
MicroChess for C64—8 levels of play	**19**
excellent graphics and color	
SPINNAKER Software C64, Apple, IBM, Atari	
Compute!'s First Book of PET/CBM	11
C64 or VIC SWITCH	**129**
POWER ROM Utilities for PET/CBM	78
WordPro 3+/64 with Spellmaster	**85**
WordPro 4+ - 8032, disk, printer	295
SPELLMASTER spelling checker ler WordPro	
VISICALC for PET, ATARI, or Apple	189
PET-TRAX PET to Epson Graphics Software	**40**
SM-KIT enhanced PET/CBM ROM Utilities	**40**
Programmers Toolkit - PET ROM Utilities	35
EASY CALC for C64	**65**
PET Spacemaker II ROM Switch	36
COPYWRITER Word Processor for C64	**69**
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET, CBM, 4040, or 8050	8
CmC Interfaces (ADA1800, AOA1450, SADI in stock)	
Programming the PET/CBM (Compute!) — R. West	**20**
Compute!'s Machine Language for Beginners	11
MULTIPLAN for C64	**79**
HES MODEM with Software	**65**
HES Software and Hardware in stock	
UMI products in stock	

FlexFile for PET/CBM/ C64 $ 50

Database, Report Writer with calculations, Mailing Lists.

FORTH for PET/C64 full FIG model — Cargill/Riley **$50**

Metacompilor for FORTH for independent object code **30**

KMMM PASCAL IV.1 for PET/CBM/C64 95

EARL for PET/CBM Disk-based ASSEMBLER **65**

Super Graphics — BASIC Language Extensions 45

Fast machine language graphics routines for PET/CBM

RAM/ROM for PET/CBM 4K $75 8K $90

DISK SPECIALS

Scotch (3M) 5" ss/dd	10/ 2.20	50/ 2.00	100/ 1.95
Scotch (3M) 5" ds/dd	10/ 3.05	50/ 2.80	100/ 2.75
Scotch (3M) 8" ss/sd	10/ 2.30	50/ 2.10	100/ 2.06
Scotch (3M) 8" ss/dd	10/ 2.85	50/ 2.70	100/ 2.65

We stock VERBATIM DISKS

Write for Dealer and OEM prices.

Sentinal 5" ss/dd	10/ 1.90	50/ 1.85	100/ 1.80
Sentinal 5" ds/dd	10/ 2.55	50/ 2.50	100/ 2.45

We stock Dysan disks

Wabash 5" ss/sd	10/ 1.60	50/ 1.55	100/ 1.45
Wabash 5" ss/dd	10/ 1.90	50/ 1.85	100/ 1.75
Wabash 8" ss/sd	10/ 2.00	50/ 1.95	100/ 1.85

We stock MAXELL DISKS

Write for dealer and OEM prices.

Disk Storage Pages 10 for $5 Hub Rings 50 for $6
Oisk Library Cases 8"—3.00 5"—2.25
Head Cleaning Kits 12
AMARAY Disk Storage Systems in stock.
Innovative Concepts FLIP 'N' FILES in stock.

CASSETTE TAPES—AGFA PE-611 PREMIUM

C-10	10/ .61	50/ .58	100/ .50
C-30	10/ .85	50/ .82	100/ .70

DATASHIELD BACKUP POWER SOURCE $265

Battery back up Uninterruptible Power Supply with surge and noise filtering. The answer to your power problems.

MultiPlan—IBM or Apple	**185**
Quadboard for IBM available	
KOALA Touch Tablets—Apple, Atari, IBM, CBM	
Peachtext 5000 Software Package	**209**
PFS Software for IBM and Apple in stock	
VOTRAX Personal Speech System	280
BMC 9191 Color Monitor	**229**
BMC 12A 12" Green Monitor	79
Brother HR-75 Daisy Wheel Printer	**769**
Dynax (Brother) DX-15 Daisy Wheel Printer	**459**
Itoh Prowriter Parallel Printer	379
Panasonic 1090 Printer with Correspondence Mode	**279**
Daisywriter 2000 with 48K buffer	1020
Gamini 10X	**299**
EPSON, Okidata, Star Micronics printers in stock	
USI CompuMOD 4 R F Modulator	29
We Stock AMDEK Monitors	
Amdek DXY-100 Plotter	590
A P Products	15% OFF
COMPUTER COVERUPS IN STOCK	
BROOKS 6 Outlet Surge Suppressor/Noise Filter	**54**
Surge Suppressor-6 outlet	29
Electrohome 1302-2 13" Hi-res RGB Monitor	335
Panasonic 12" Monitor (20 MHz) with audio	135
Synertek SYM-1 Microcomputer	**189**

ALL BOOK and SOFTWARE PRICES DISCOUNTED

USI Video Monitors—Green or AMBER 20 MHz hi-res.
Dealer and OEM inquiries invited

ZENITH | data systems

ZVM-122A	109	ZVM-123G	95
ZVM-131	300	ZVM-135	510
HERO 1 Robot (factory assembled)			2145
Z29 Terminal (DEC and ADM compatible)			680
ZT-10 Intel. Terminal with Serial Port			340
Z100 16-bit/8-bit Systems			CALL
We stock entire Zenith line.			

ATARI — WE STOCK ENTIRE LINE

SPINNAKER and Broderbund Software in Stock.

215-822-7727
252 Bethlehem Pike
Colmar, PA 18915

A B Computers

WRITE FOR CATALOG. Add $1.50 per order for United Parcel. We pay balance of UPS surface shipping charges on all prepaid orders (add extra for mail, APO/FPO, air). Prices include cash discount. Regular prices slightly higher. Prices subject to change.

Manufacturer: Koala Technologies
253 Martens Ave.
Mt. View, CA 94040
Author: Steven Dompier

Description: A graphics illustrating program with various brush sets, colors, and drawing modes. Points, lines, rectangles and circles can be plotted on the high resolution screen with different colors. Filling and erasing is accomplished easily. Creations can be saved to and loaded from disk. The program is a combination of BASIC and machine-language on an unprotected disk.

Pluses: The program is menu driven and much easier to use than most graphics programs. It includes many sophisticated features such as magnification and fill. It is very easy to create complicated and professional looking drawings.

Minuses: The Koala Tablet is recommended for its smoothness and preciseness. It is difficult to draw a smooth line with a joystick.

Documentation: A very complete, clear and well written booklet is included with the program.

Skill level required: No prior skill needed. Good for ages 5 and up.

Reviewer: Phil Daley

Product Name: **Bugbyter**
Equip. req'd: Apple II
Price: $47.50
Manufacturer: Computer-Advanced Ideas
1442A Walnut Street, Suite 341
Berkeley, CA 94709
Author: Tom Cohn, and Pete Row

Description: *Bugbyter* is an interactive M/L debugging tool which is relocatable to any unused location compatable with your program including into a language card. It has a screen-oriented display which has complete information on the status of all the 6502's registers and stack. It includes capabilities to turn off the display to avoid conflict with the program. and to set an area of memory (such as DOS or the monitor) to be executed at real-time known code. You can set breakpoints-hard or soft-and triggers to count the break points. It assembles and disassembles code similar to and uses instructions like the monitor. The accompanying 40 page documentation is excellent.

Pluses: It is very easy to use and clearly shows everything you need for debugging. No M/L programmer who writes large programs should be without it.

Minuses: None.

Skill level required: Beginning to intermediate M/L programmer.

Reviewer: Phil Daley

MICRO

SOFTWARE ENHANCEMENT SYSTEM, APB-102 ..... $189.00
W/GPLE, APU-1, FMP, DISK, MANUAL, QUICK REFERENCE GUIDE
APU-2, UTILITY ROM #2 ..... $35.00
W/RENUMBER, MERGE, HOLD, ETC.
ROM DEVELOPMENT PKG. $99.00
W/DISK, INSRUCTIONS & EMULATION ROM
A/D, 12 BIT, 16 CHANNEL $299.00
W/5 VOLTAGE RANGES, 25 μ SEC CONVERSION
PRO-1, XTRA-LARGE PROTOTYPE BOARD $29.95
UP TO 52 IC's, NUMBERED & LETTERED PINS, HANDY POWER AND GROUND CONNECTIONS, NUMBERED I/O CONNECTIONS
48 LINE PARALLEL I/O BOARD, CPU-1 ..... $249.00
25 BUFFERED LINES IN W/FILTERING, 23 BUFFERED LINES OUT, INTERRUPT INPUT

#1 SELLER AT A.P.P.L.E.*

Powerful New Software Enhancement System For Apple II, //e. Triples Programming Speed!

APB-102 Ultra-Rom Board/Editor Includes:

- Advanced G.P.L.E.* (Global Program Line Editor) in Firmware
  —With Insert, Delete, Find, Tab, Zap, Pack, Restore, End, Etc.
- Firmware Management Program (FMP) Overlays 32K of ROM in 2K Space
  —Allows jumps and calls between banks—searches for utilities by name
  —Recognizes new ROMs and utilities automatically
- APU-1 with over 25 Language Extensions & Ampersand Utilities
  —If/Then/Else, Print Using, Ultra Fast Search, Damaged Program Recovery
- Always in the Machine—No Searching for a Disk
- Never in the way—No Program RAM used
  —Connect with 4 Keystrokes / Disconnect with 2!

"If you program & haven't used a line editor, get one right away" —MICRO MAGAZINE
"The most powerful program development tool I have" —ROBERT WILSON, PROGRAMMER
"Excellent Product, flawless" —PHILIP DALEY, PROGRAMMER
"Great product, exceeds my expectations" —DR. STEVE COOK
"An elegant solution... well thought out... worth it" —SOFTALK REVIEW, SEPT. 1983
"The best thing for the Apple since the disk" —EDWARD DECKER, PHARMACIST/PROGRAMMER

Hollywood Hardware (213) 989-1204
6842 VALJEAN AVENUE, VAN NUYS, CA 91406

USE OUR 60 DAY UNCONDITIONAL MONEY BACK TRIAL.
IF YOU CAN PART WITH IT—WE'LL BUY IT BACK!
ASK ABOUT DEALER/USER GROUP DISCOUNTS!

VISA MasterCard

*G.P.L.E. © 1983 NEIL KONZEN. SOLD UNDER LICENSE FROM SYNERGISTIC SOFTWARE / APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTER INC. A.P.P.L.E. IS APPLE PUGETSOUND PROGRAM LIBRARY EXCHANGE, THE WORLD'S LARGEST APPLE USERS' GROUP WITH 25,000 MEMBERS.

SJB DISTRIBUTORS. ONE STOP SHOPPING FOR COMMODORE SYSTEMS.

commodore

NEW COMMODORE PRODUCTS

Executive 64	$ Call
CBM B128-80	825
B Series Software	Call

WORD PROCESSING 64

NEW - Mirage 80 col	$ 95
WordPro 3+/Spellright	79
WordPro 3+ (WP)	59
Spellright (Dictionary)	39
Paper Clip	95
Easy Script (D)	35
Easy Spell (D)	19
SPECIAL - Busiwriter (C,D)	39
Quick Brown Fox (R)	49

SPREADSHEETS 64

Calc Result - Advanced (R,D)	$ 120
Busicalc II - More Power! (R,D)	95
Multiplan 64 (D)	79
Calc Result - Easy (R)	70
Practicalc 64 (D)	45
Busicalc I - SPECIAL! (C,D)	39

DATA BASES 64

Mirage Data Base (D)	$ 95
M'File (merges with WordPro) (D)	89
Micro Spec Data Manager (D)	60
Codewriter (develops programs)(D)	95

PRODUCTS OF THE MONTH

TCS 64/80 - NEW! (D) (WP/Data Base/Spread)	$ 139
Koala Pad - NEW!	89
Delta 10 (160 cps) - NEW!	549

UTILITIES 64

Vic Tree (4.0 Basic) (R)	$ 75
64 Super Expander (R)	25
Simon's Basic (R)	25
Cardco Printer Utility (C)	15
MS-Backup (Back Up Data!) (D)	15

ACCOUNTING 64

Home Accountant (Continental)	$ 75
Tax Advantage (merge w/home accountant) - NEW!	45
General Ledger, A/R, A/P, P/R, Inv. (Info Design's Original) (D)	ea.79
Numeric Keypad (Hardware)	65
Numeric Keypad (Cardco)	35

TELECOMPUTING 64

Vic 1650 (Auto Ans/Dial) Modem	$ 95
Vic 1600 Modem	59
Hes Modem (Downloading Software)	65
Super Term (Download/80-128 Form).	95
Micro Term 64 (Download P/D)	39

EDUCATION 64

Spelling I (Koala) (D)	$ 29
Geometric (Koala) (D)	20
I.Q. Baseball (D)	25
Bible Baseball (D)	25
Happy Tutor (Typing) (D)	15

LANGUAGES 64

Instaspeed Basic Compiler (D)	$ 99
Nevada Cobol (D)	55
Pilot (D)	45
Logo (D)	45
Assembler Development (D)	25
64 Forth (R)	40

PRINTERS - DOT MATRIX

Epson RX80 (80 cps)	$ 299
MX80 w/FT (80 cps)	399
FX80 (160 cps)	549
FX100 (160 cps) 14" width	859
Okidata 82A	429
Okidata 92	549
NEC 8023A	429
Star Delta (160 cps)-NEW!	549
Star Gemini 10X (120 cps)	309
Star Gemini 10/15	Call
Transtar 315 (Hi Res., Color)	575
Micro Edge Printer Paper (540 Sheets)	10

ESSENTIALS

Commodore 64	$ Call
1541 Disk Drive	249
1525 Printer (80 col/DM)	225
1530 Datasette	65
1520 Plotter/Printer (4 Color)	169
1526 Printer	Call
1702 Monitor	249

CBM

8032 (80 column Pet)	$ 625
SuperPet (5 languages!)	1049
8050 Dual Drive (1 mg.)	995
8250 Dual Drive (2 mg.)	1295
9060 Hord Disk (5 mg.)	1995
2031, 170K Single Drive	295
64K Upgrade for 8032	259
SuperPet upgrade for 8032	599
4023 Printer (80 cps, 80 col)	395
8023 Printer (150 cps, graphics)	545
6400 Printer (40 cps, LQ)	1450

LETTER QUALITY PRINTERS

Diablo 620, 25 cps	$ 949
Transtar 130, 16 cps - 132 col	769
Transtar 120, 14 cps - 80 col	500

MONITORS

Panasonic CT 160 (color)	$ 279
Panasonic TR120 (w/spkr,green)	155
Sanyo/Amdek-Green, No Audio, 12"	125
BMC/Sanyo-Green, No Audio, 9"	95
Cable (For Above) A/V	15

VIC ACCESSORIES

8K RAM Expand. Cart	$ 40
16K RAM Expand. Cart	70
24K RAM Expand. Cart	105
27K RAM (Expands Vic to full 32K)	119
3 Slot Expander	30
6 Slot Expander	70
Joystick (Wico-Red)	25
Joystick Blaster (ADR Rapid-Fire)	10

INTERFACES & ACCESSORIES

Data 20 80-Col. Exp.	$ 159
Mr. Computer 80-Col. Exp.	60
5-Slot Exp. (64)	65
Vic Switch (connect 8 64's or Vic's to DD/Printer)	145
Cables 3M, 6M, 12M for above	Call
Verex (Box of 10) 5¼ Diskettes	26
Connection(Pet/64 graphics,2K Buffer)	99
Cardco Print + Graphics	85
Cardco Cardprint	70
MW 302 Parallel	65
PET/IEEE Cable (1m)	33
IEEE/IEEE Cable (1m)	49
Interpod (Intelligent IEEE, RS232, serial)	149
ADA 1800 (IEEE/Parallel)	129
ADA 1450 (IEEE/RS232 (M/F))	129

VISA/MASTERCARD
MONEY ORDERS
BANK CHECK

C.O.D.'s Accepted. (Add $5)
In stock items shipped within 48 hours.
F.O.B. Dallas, Texas (Texas Res., Add 5% Tax).
Products shipped with manufacturer's warranty.
Prices subject to change without notice.
$50 Minimum Order.

*Defective units **must have** return authorization number and include copy of invoice.

SJB DISTRIBUTORS INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238

TO ORDER
CALL TOLL FREE
800-527-4893
800-422-1048
(Within Texas)

CATALOG
Send Postcard with Name & Address to speed processing.

NOTE: SJB HAS A FULL LINE OF COMPUTER MEDIA IN STOCK, CALL OR WRITE FOR MORE INFORMATION.

MICRO™

Interface Clinic

by Ralph Tenney

Dave English, of Orange, California wrote with the following question: "I have never seen a general treatment of direct interfacing between micro CPUs, so that one runs foreground and one runs background. This does not seem to be a difficult topic, as there are a variety of plugin modules are sold for various computers and other microcomputers with dual CPUs, such as the SuperPET."

Dave then goes on to mention one of his own experiments: "I currently have a Timex 1000 and a PET 2001 sharing the screen of the PET. It's not true time sharing; one or the other has complete control of the screen at any time."

We can address part of Dave's question by examining the factors involved in getting two (or more) CPUs to cooperate in various ways. The first thing is to define the extent of the interaction and decide what type of interface is needed. We also need to understand that not all of the plugin boards sold for computers such as Apple are multi-processor setups, even though they may seem to be. Many of the popular "appliance" computers have a special input line in their expansion port to disable the processor so that an external plugin can take command of the memory. So, as long as that plugin is in place, only the processor on the plugin is active. I think it is fair to say that this setup doesn't really fit Dave's topic.

There are two basic types of multi-CPU interfaces — one is where the various CPUs communicate directly on a shared bus, and, strange as it may seem, this can be done even with dissimilar processors. It is perhaps more convenient to do both the programming and the interface design if the shared-bus concept uses identical processors, but mutual operation on a shared bus is possible with different processors. The second mode is one where the necessary communication is carried on an auxiliary bus such as an RS-232 link or a high speed synchronous or asynchronous line. This type of communication can also be performed by similar processors (such as two 6502s) or different processors (such as 6809 and Z-80).

During this discussion we will ignore those microprocessor families which have coprocessors available; a coprocessor is a specially programmed subset of the main processor, dedicated to a special task such as number crunching or I/O. These are a special case, and the interaction has been carefully planned by the manufacturer. We will also not deal with the concept of multi-

NO POWER SPIKES WITH SUPER FAN II.

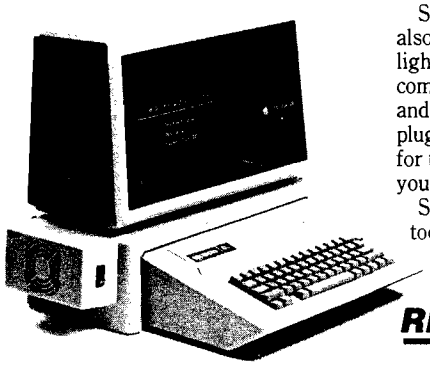
Super Fan II's Zener Ray™ Transient Voltage Suppressor and Power Filter squelches spikes up to 6000 amps—even those caused by lightning—while responding up to 100 times faster than Apple II's built-in suppressor.

In addition, Super Fan II cools your Apple, removing heat buildup at a remarkable 17 cubic feet of air per minute. Yet it's the quietest fan of its kind on the market.

AIR CONDITIONED

Super Fan II also positions a lighted on/off computer switch and two accessory plugs at your fingertips. It's warranted for two years and simply clips to your Apple II, IIe or monitor stand.

See your R.H. Electronics dealer today about Super Fan II*, or contact us at 566 Irelan Street, Buellton, CA 93427, (805) 688-2047.



RHELECTRONICS, INC.

Super Fan II, in black or tan. $109. Without Zener Ray, $74.95. Additional air flow seals, $5. Available in 240V/50 Hz.

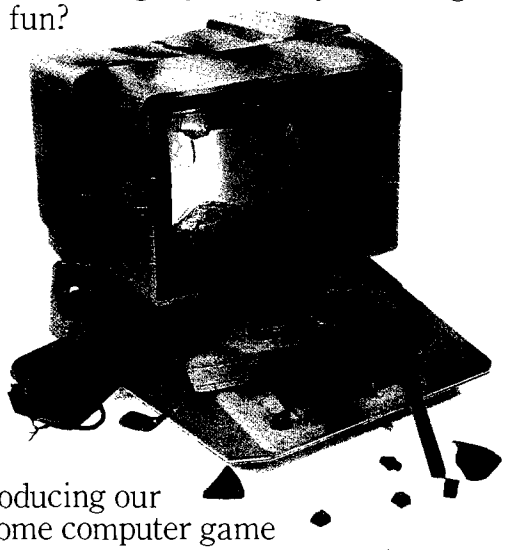
*Dealer/OEM inquiries invited. *U.S. Patent #D268283 #4383286*

France, call B.I.P. 1-255-4463 Australia, call Imagineering (02)212-1411

Blow up

Gee whiz. Look what you've done. Your big shiny Apple, destroyed.

And you thought you were just having a little fun?



Introducing our new home computer game Cavern Creatures.™ Where one false move, one mistaken twitch of the Joystick can do terrible things to an Apple. Like shut parts down.

Who knows which ones? Making it act awful funny and then... KA-BOOM!

The game's finished. And maybe, so is your Apple. Oh, but don't worry.

Cavern Creatures is just a game. Full of action. Fun. Suspense. Just like our other games. Whether they're blowing up your Apple. Or totally blowing your mind.

How Much More Can One Joystick Jockey Take?

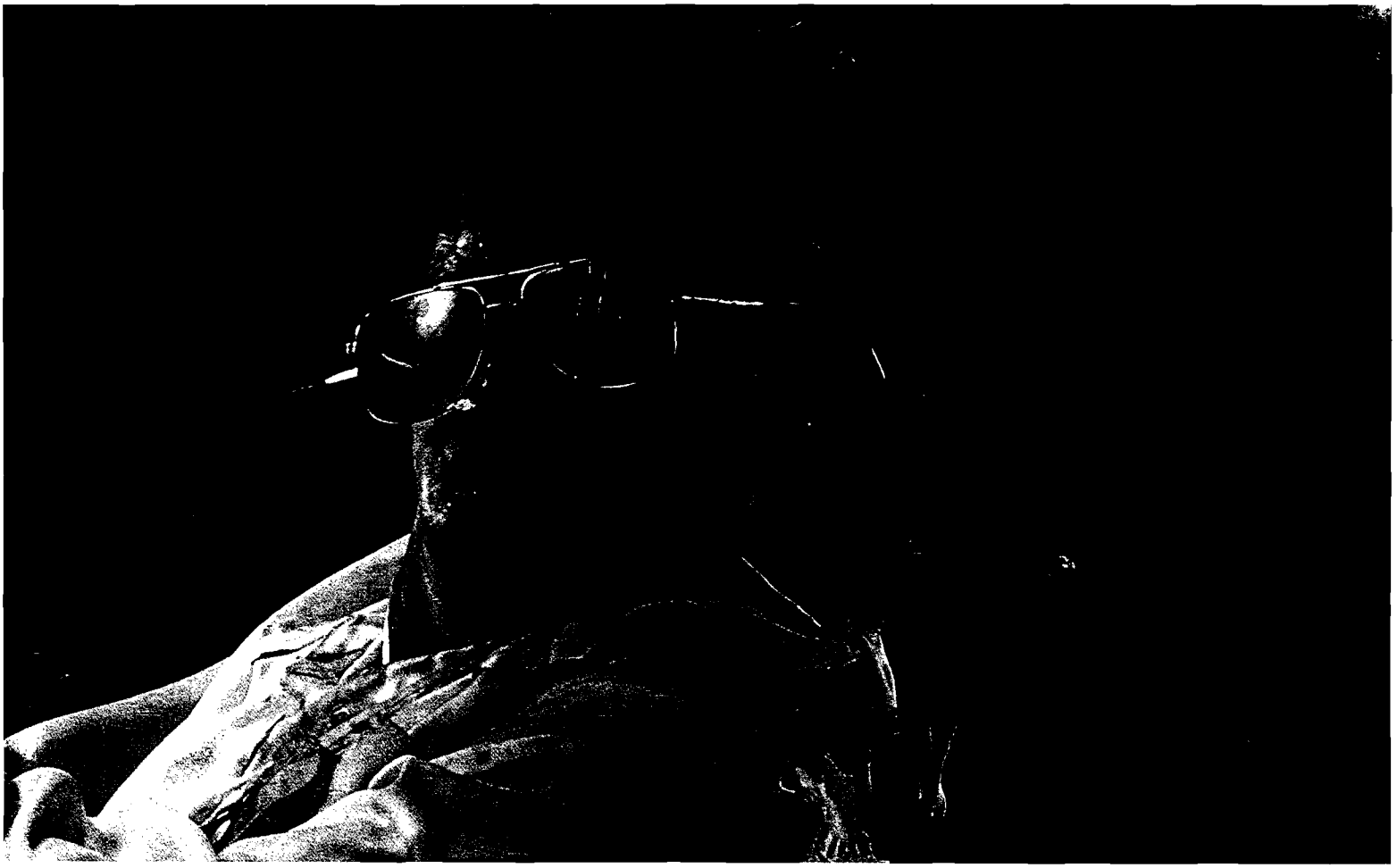
We've exceeded our past reputation for bringing you some of the most super snazzy, ultra-intense, graphically involving games around.

We've gone totally out of our minds with every game from shoot 'em up, blast 'em out of the sky strategies to mind game graphics guaranteed to provoke a mental meltdown.

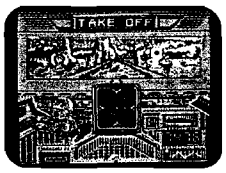
Are you ready?

Cavern Creatures™

You can squirm. Beg. Plead. And moan. Nothing will save you, or your Apple, from being blown sky-high by this game. Don't worry. The effect's not permanent. Your Apple will recover. But will you?

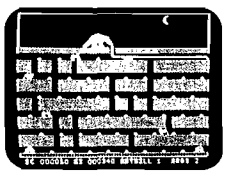


your Apple.*



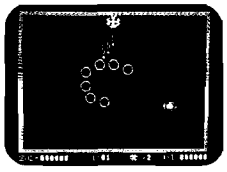
Space Ark™

Skin-hunters, poachers and mechanicals—enemy robots who aren't exactly the warm and friendly types—await you. Obliterate the bad guys. Save the good guys. You might survive this.



Ardy The Aardvark™

Here's a tongue that's as long and skilled as you are. Think you can lick stinging ants and tarantulas? Use Ardy's tongue as you make it through this maze!

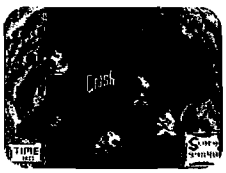


Roundabout™

Sharpens your target skills with 24, count 'em, 24 different series of targets. How's your hand to eye coordination, pal?

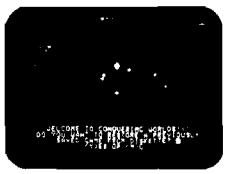
Argos™

And if you like to push people around, lay off your little brother and take on these aliens instead. Save the Domed City from doomsville.



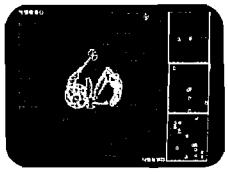
Super Bunny™

No, *Super Bunny* is not referring to all those Playboys stashed under the bed. It's a strategy/action game that just might turn your brains into carrot puree. Cover your burrow!



Conquering Worlds™

So you fancy yourself a die-hard strategist? Try conquering this. Maybe you'll rule the universe if mom will let you off restriction.



Bilestoad™

Avoid violence, but be prepared. Incredible graphics make this rated R! But if you're a wimp, pass this game up. It's not for babies.

DATAMOST

The most out of our minds.™

TM is a registered trademark of Datamost, Inc. 8943 Fullbright Ave., Chatsworth, CA 91311
(213) 709-1202 *Apple is a trademark of Apple Computers. © Datamost 1983

ple processors for the purpose of reliability (redundant design). Instead, I will give some points to consider and hints on how to proceed with designing a system of multiple processors which share the system resources but each "does its own thing". One example is a commercially available dedicated word processor. In this particular system, a 6800 handles all the word processing while a 6502 runs the keyboard, screen display and the disk drives. This particular machine can receive serial data at 1200 baud, storing the incoming data on the disk and displaying it at the same time. The major system requirement for multi-processing is to partition all the jobs, defining very carefully the responsibility of each processor. In fact, it is like designing two separate systems and then designing an interface between the systems! Once all the system jobs have been passed out, you need to define the interface and decide how the information interchange will be handled.

Some microprocessors have been designed specifically for multi-processing, such as the 6801 and 6809. Figure 1 shows just the interconnection of two 6801 uPs. The eight lines of Port P3 and the two SC lines combine to make a parallel link which transfers one byte at a time between the two devices. The SC lines form a handshake setup which informs the "slave" processor that data is ready. Note that "slave" and "master" status is defined by which is programmed to send and which is programmed to receive. The 6809 has a special pin called DMA/BREQ which allows other processors or special purpose hardware to request control of the data and address bus lines for fast data transfer or other purposes. The Radio Shack Color Computer uses the 6809E, which does not allow normal DMA or other types of bus access, since this would disturb the special relationship between the uP and the SAM (Synchronous Address Multiplier).

Another way to interface between processors is to define special areas of memory which can contain anything from a single bit or byte to whole pages of data. One processor must leave the data for the other, and set a flag which indicates the data is ready. The second processor will then pick up the data and cancel the flag. In practice, the memory bus must be segmented so that each processor has enough "private" memory for its own tasks, and the data buffer must be in address space which each processor can use. At the same

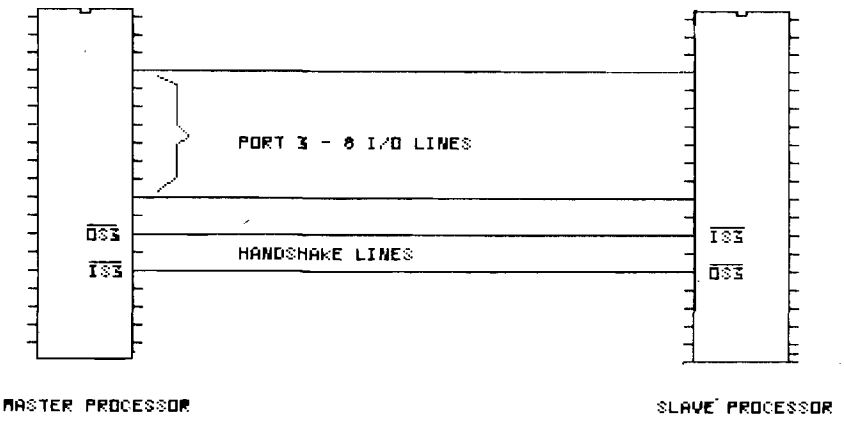


Figure 1: **Some microprocessors, such as the Motorola 6802 shown here, have built-in provisions for communication between processors.**

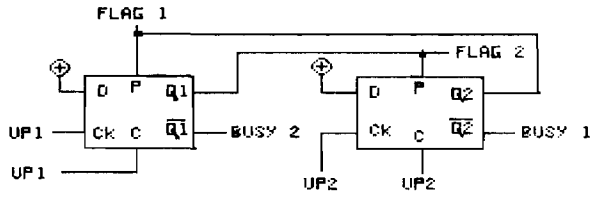


Figure 2: **Two processors sharing some common memory area can use interlocked latches to prevent bus conflict (see text).**

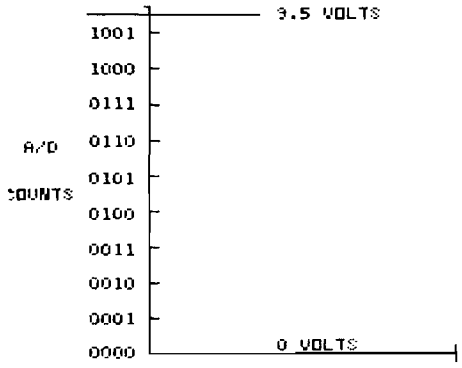


Figure 3: **A Digital-to-Analog converter cannot make accurate measurements without adequate resolution (see text).**

time, there needs to be either interlock circuitry which prevents both processors from trying to access the area simultaneously, or the programming for both processors must enforce a similar protocol. This type of design almost always involves a dedicated pair of latches, and each latch will have a unique address controlled by one processor. This is illustrated in Figure 2; when uP#1 is using the data buffer, it sets latch L1 to the "off" state (Q = 0) by pulsing the CLEAR input. L1's Q output holds L2 in PRESET (Q high) as long as uP#1 is accessing the buffer. At the same time, L1's /Q output is high, serving as a BUSY flag for uP#2. When uP#1 releases the buffer, it pulses the clock line of L1, which sets L1 to the "on" state, clears the busy flag and releases L2. Similar action by uP#2 controls L2 and warns uP#1 that the buffer is busy. It may also be important that the two processors have synchronized clock inputs.

If two separate computers involved in a project must work together, one computer can be designated as the master, and it will send commands to the other computer over their link, which can be a high speed RS-232 line. The master must be programmed to pass the required commands to the slave computer, besides keeping its own business straight. It is likely that the slave will be assigned some kind of monitoring tasks, and the incoming commands will trigger interrupts on the slave. Then, depending upon the type of command, the slave may make a different measurement or report the previously collected data. Whatever the assigned tasks, part of the programming for both computers will involve specially coded commands to define precisely the nature of the task to be performed.

That's enough on computer communications for now! I have promised to introduce some of the real-world hardware such as analog to digital (A/D) and digital to analog (D/A) converters. Very few processes and parameters we might be interested in are digital. We normally want to know how heavy, how strong, how loud, etc. something is. Natural processes come in all sizes, shapes and colors, and a precise measurement will usually involve values which do not change in discrete digital steps.

Natural processes vary widely between two limits in such a way that a digital quantity will not be exactly correct as shown in Figure 4. This diagram assumes an A/D converter with a full scale input of 16 volts and having a resolution of four bits (2**4, or one part in 16). In other words, each step of the converter represents a one volt change. If we assume the voltage to be measured is 9.5 volts, we find that step 1001 (binary) is too low and step 1010 is too high. Note that even if the 4 bit converter is 100% accurate, the best answer it can give for this measurement is incorrect by .5/9 or 5.5%. If we can increase the converter's resolution to 8 bits (2**8, or 1 part in 256), 9.5 volts happens to be exactly 152 steps out of the 256 possible; the new converter has steps of 62.5 millivolts. At present, about the best resolution economically possible with an A/D converter is 16 bits (one part in 65536), or 244 microvolts if the full scale range were 16 volts as in our examples. Actually, typical A/D converter ranges are 0 - 5 volts, 0 - 10 volts, or -5 to +5 volts. If we don't have another reader question next month, we will get into types of A/D converters and do some experiments.

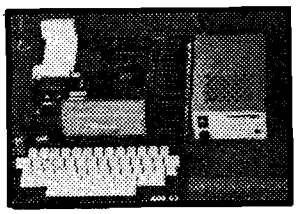
You may contact Mr. Tenny at P.O. Box 545, Richardson, TX 75080.

MICRO

AIM + POWER

from **COMPUTECH**

All prices **Postpaid** (Continental U.S. — otherwise $2 credit)



Check the **outstanding documentation** supplied with AIM65

Top quality power supply designed to Rockwell's specs for fully populated AIM 65 — includes overvoltage protection, transient suppression, metal case and power cable:

PSSBC-A (5V 2A Reg; 24V .5A Avg, 2.5A Peak, Unreg) ...**$64.95**

Same but an extra AMP at 5 volts to drive your extra boards:

PSSBC-3 (5V 3A Reg; 24V .5A Avg, 2.5A Peak, unreg) ...**$74.95**

The **professional's** choice in microcomputers:

AIM65/1K RAM.....**$429.95** BASIC (2 ROMS) **$59.95**

AIM65/4K RAM.....**$464.95** ASSEMBLER (1 ROM) ..**$32.95**

FORTH (2 ROMS) .....**$59.95.**

SAVE EVEN MORE ON COMBINATIONS

AIM65/1K + PSSBC-A .**$479.95** AIM65/4K + PSSBC-3 .**$524.95**

We gladly quote on all AIM65/40 and RM65 items as well.

ORDERS: **(714) 369-1084**

P.O. Box 20054 • Riverside, CA 92516

California residents add 6% sales tax

LOOK TO
MACMILLAN
FOR COMPUTER SCIENCE

THIS IS VIC-20 BASIC

ROBERT F. SUTHERLAND
and RICHARD M. GILMAN

Emphasizing good programming practices, the authors show how to take advantage of the VIC-20's special capabilities such as creation of shapes and pictures in color, and production of music and sound effects.

1984 Paperback 420 pp.(approx.) ISBN: 0-02-418380-6 $16.95

AN INTRODUCTION TO STRUCTURED BASIC FOR THE CROMEMCO C-10

WAYNE T. WATSON

The only book available in its area, this book assumes no prior programming experience and stresses interaction with the microcomputer. The first ten chapters provide a foundation to the most commonly used concepts in BASIC, and the last four chapters offer more advanced material on files and programming structures.

1984 Paperback 276 pp.(approx.) ISBN: 0-02-424580-1 $16.95

BASIC-80 AND CP/M

JACK JAY PURDUM

The key features of BASIC-80© are related to the most popular microprocessor operating system, CP/M.™ Coverage includes sequential, random, and skip-sequential file structures and many useful subroutines for applied programs (binary searches; Shell and Bubble sort; range checks; direct cursor control for many popular CRT's; and error messages under direct cursor control).

1984 Paperback 220 pp. ISBN: 0-02-397020-0 $16.95

IBM-PC: 8088 MACRO ASSEMBLER PROGRAMMING

DAN ROLLINS, Independent Consultant

Intended specifically for the increasingly popular IBM-PC, this book incorporates up-to-date sections on IBM-PC color graphics and includes many sample programs.

March, 1984 Paperback 300 pp. ISBN: 0-02-403210-7 $13.95–tentative

Prices subject to change without notice.

AVAILABLE AT YOUR LOCAL BOOKSTORE

FOR MORE INFORMATION CALL TOLL-FREE 1-800-223-3215. *OR WRITE:*

MACMILLAN PUBLISHING COMPANY
College Division • 866 Third Avenue • New York, NY 10022

Advertiser Index

A B Computers	88
Amplify	36
Atari Program Exchange	6
CompuTech	94
Computer Mail Order	76,77
Datamost, Inc	92,93
Eastern House Software	13
Hollywood Hardware	89
Incomm	42
Interesting Software	38
Kiwi Software	73
Leading Edge	Back Cover
MacMillan Publishing Co	96
Micro	75, 80
Micro Spec	9
Micro Ware	45
Midwest Micro Inc	22
Percom Data	3
Performance Micro Products	59
Perry Peripherals	72
Protecto Enterprizes	30, 31, 32
RH Electronics, Inc	91
Richvale Telecommunications	Inside Front
S G C	86
S J B Distributors	90
Safeware	56
Scientific Software	73
Skyles Electric Works	19
Spectrum Projects	70
Star Micronics	Inside Back Cover
Strom Systems Inc	2
Titan Technogies, Inc	20
Totl Software	58
Victory Software	21
Winders & Geist, Inc	1

National Advertising Representatives

Home Office:
Cindy Kocher, Advertising Manager
P.O. Box 6502
Chelmsford, MA 01824 (617) 256-3649

Mid-West Territory:
Thomas Knorr & Associates
Thomas H. Knorr, Jr.
333 N. Michigan Avenue, Suite 401
Chicago, Illinois 60601 (312) 726-2633

serving: Ohio, Oklahoma, Arkansas, Texas, North Dakota, South Dakota, Nebraska, Kansas, Missouri, Indiana, Illinois, Iowa, Michigan, Wisconsin, and Minnesota.

West Coast:
The R.W. Walker Co., Inc.
Gordon Carnie
2716 Ocean Park Boulevard, Suite 1010,
Santa Monica, California 90405 (213) 450-9001

serving: Washington, Oregon, Idaho, Montana, Wyoming, Colorado, New Mexico, Arizona, Utah, Nevada, California, Alaska, and Hawaii (also British Columbia and Alberta, Canada)

The new pacesetter in professional printers.

The economical dual-speed Radix-15 multi-function printer.

With a fast and furious work pace, a highly flexible printer is crucial. That printer is the new Radix-15. Watch it take your work and run with it.

It's dual speed! At 200 cps Radix fires out a superbly refined dot matrix printout. At 50 cps it prints professional near-letter quality. So now you can go from spreadsheets to memos at the flip of a switch or at your computer's command.

It's multi-functional! In either mode Radix-15 quickly adapts to your needs. There's serial and parallel interface. Memory storage with a 16K buffer. Responsive throughput to help you use time more efficiently. Bi-directional performance. Friction and tractor feed. An automatic sheet feeder for letterheads and a short form tear-off for preprinted forms. Plus, the freedom to underline, set vertical and horizontal tabs and print a huge variety of type faces.

It's economical! And not only does Radix give you 2 printers in 1, it also gives you a price performance as outstanding as its working performance.

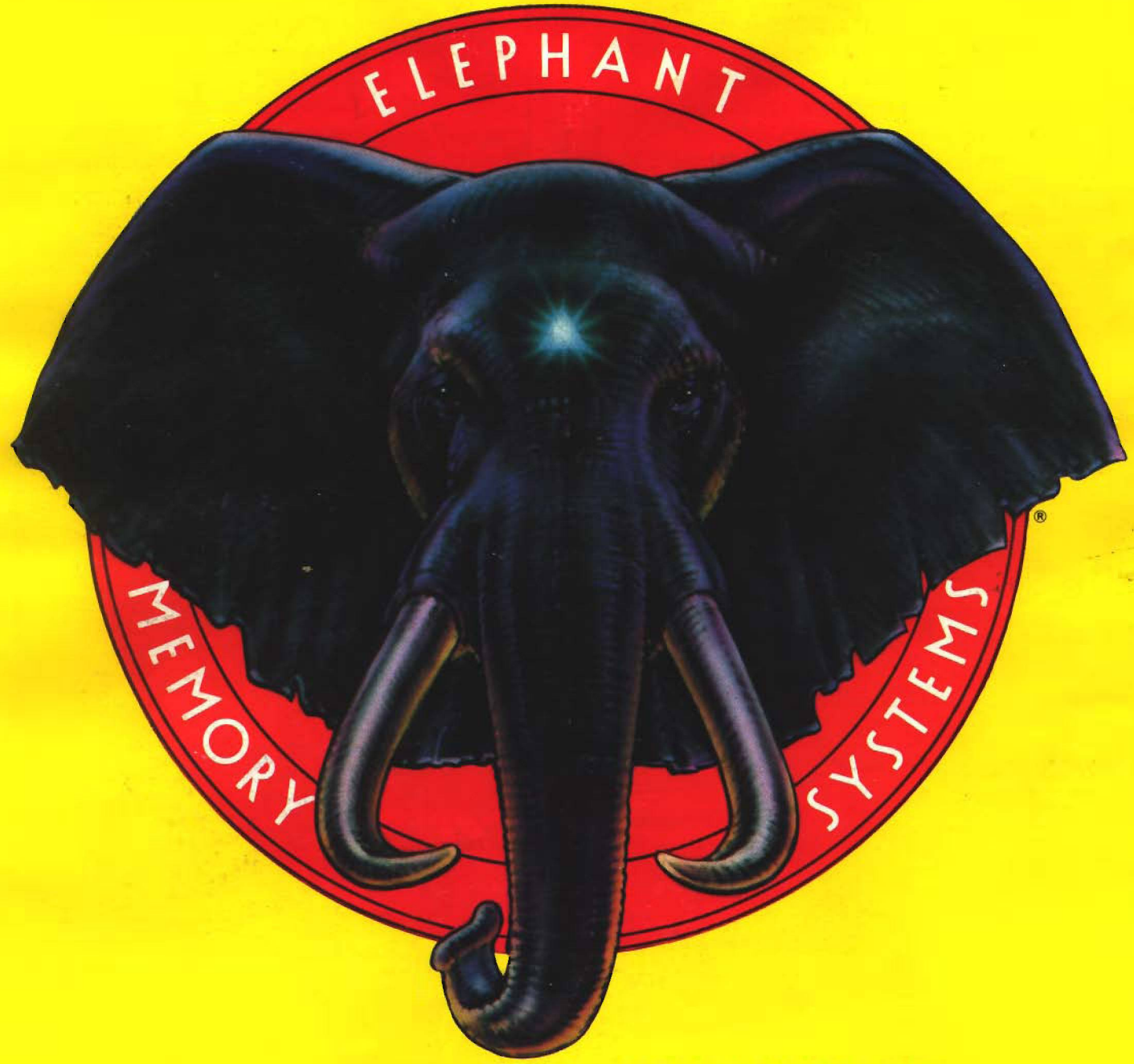
The professional Radix-15. Using the ever-changing beat of business to your advantage. And that's what being a pacesetter is all about!

star™ micronics·inc

THE POWER BEHIND THE PRINTED WORD.

Computer Peripherals Division
P.O. Box 612186, Dallas/Ft. Worth Airport, TX 75261 (214) 456-0052

REMEMBER.



ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison